IPP-Report

IPP Max-Planck-Institut für Plasmaphysik

Mutzke, A., Schneider, R., Eckstein, W., Dohmen, R.; Schmid, K., von Toussaint, U., Bandelow, G.

SDTrimSP Version 6.00

IPP 2019-02
Februar 2019

SDTrimSP Version 6.00

A. Mutzke[a], R. Schneider[b], W. Eckstein[a], R. Dohmen[a],
K. Schmid[c], U. von Toussaint[d], G. Bandelow[e]

[a] Max-Planck-Institut für Plasmaphysik, Garching, Germany
[b] Universität Greifswald, Greifswald, Germany
[c] chapter. 7.4: Max-Planck-Institut für Plasmaphysik, Garching, Germany
[d] chapter. 7.5: Max-Planck-Institut für Plasmaphysik, Garching, Germany
[e] chapter. 7.6: Universität Greifswald, Greifswald, Germany

Abstract

SDTrimSP version 6.00 simulates designed for atomic collisions in amorphous targets. It calculates ranges, reflection coefficients and sputtering yields as well as more detailed information like depth distributions of implanted and energy distributions of backscattered and sputtered atoms. The program is based on the binary collision approximation and uses the same physics as its predecessors TRIM.SP and TRIDYN, but the structure of the new program has been completely changed. It runs on all sequential and parallel platforms with a F90 compiler. Table lookup is applied for all available atomic data needed for input, and different integration schemes for several interaction potentials are provided. Several examples are given to show the wide range of possible applications.

This report replaces the previous version (IPP 12/8, 2011).

Contents

1. Introduction

In the last 50 years many computer simulation programs have been developed to describe the interactions of ions bombarding solid, liquid and gaseous targets. Many of these programs were based on the binary-collision approximation dealing with crystalline and amorphous targets. Examples of programs dealing with amorphous targets are the static Monte-Carlo program TRIM and the corresponding dynamic version TRIDYN which were successful in describing collision effects in solids for many examples [1]. Many versions of these two programs evolved to handle specific physical problems. This triggered the idea to combine both programs in a new version SDTrimSP (where S stands for static and D for dynamic) with all possible output capabilities used in the past like sputtering, backscattering and transmission. This offered the opportunity to introduce at the same time a modular structure, to have a more flexible output and to provide a higher portability. The program is suited equally well for all sequential architectures and for all parallel architectures, for which a Fortran 90 (F90) compiler and the MPI (Message Passing Interface) communication library are available. The new program also includes features which were used in the past, but were not incorporated in most versions of TRIM.SP and TRIDYN.

The extensions for SDTrimSP version 6.00 are described. The main new features are the inclusion of thermal-diffusion and new modules of surface-binding and inelastic energy-loss. The models of out-gasing for noble gas ions and chemical erosion of carbon have been improved and adapted.
Outlook:Based on the principles of SDTrimSP 6.00 also code versions for 2D- and 3D-problems have been derived [56],[57] , which however are described in separate upcoming reports.

2. Physical basis

The new program SDTrimSP is based on TRIM.SP [2] and TRIDYN [3, 4]. Both programs, the static TRIM.SP and the dynamic TRIDYN, are described in [1]. The basic physics in the program SDTrimSP is the same as in the former versions. SDTrimSP is a Monte Carlo program, which assumes an amorphous (randomized) target structure at zero temperature (ie. static target atoms $|\vec{v}| = 0$) and infinite lateral size.
The target in SDTrimSP is one dimensional (Fig. 1). The target is divided into layers. Y and Z direction are taken as infinite. In the static mode the target is fixed. In the dynamical mode the thickness of layers is changed, [26].

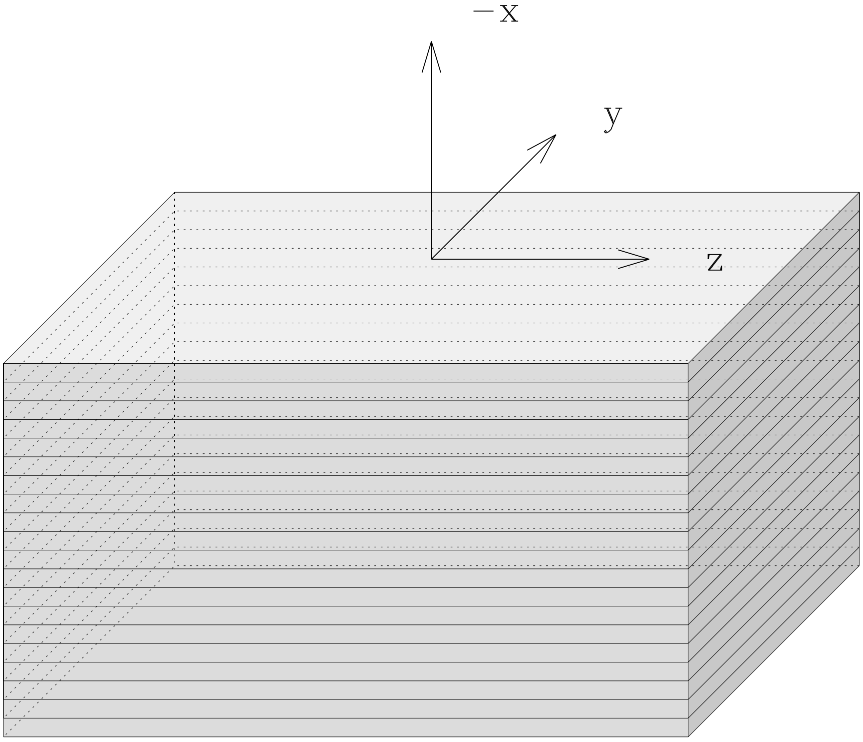

Figure 1: Geometry of the one dimensional target

The binary collision approximation is used to handle the atomic (nuclear) collisions. This means, that the change in flight direction due to the collision is given by the asymptotes of the real trajectory.

For this evaluation an interaction potential has to be chosen (usually purely repulsive and only dependent on the distance between the colliding atoms) to determine the scattering angle of the moving atom and the recoil angle of the atom, which is set into motion. Then the energy loss (nuclear) of the moving atom and the energy gain of the recoil can be calculated. In addition, a moving atom looses energy to target electrons (electronic or inelastic energy loss). The program also provides the possibility to include simultaneous weak collisions, but strictly in the binary collision approximation. The program follows projectiles (incident atoms) and target recoil atoms three-dimensionally until their energy falls below some preset value or if they have left the target (backscattering, transmission, sputtering). Besides a more modular structure many new features are included in the program. Most data needed for a calculation is taken from a database in form of tables: atomic numbers and masses of elements, densities of solid and liquid elements, surface binding energies (heat of sublimation). Displacement energies are taken from Table 6.1 of [1]; one table provides isotopic masses of elements; two other tables give the constants for the inelastic energy loss of hydrogen [5] and helium [6]. Different interaction potentials as KrC [7], ZBL [8], Molière [9], Nakagawa-Yamamura [19] , power potentials and a special Si-Si potential [10] can be chosen as well as different integration methods of the scattering integral as Magic [11], Gauß-Mehler [12], and Gauß-Legendre [13]. Magic is faster than the Gauß-Mehler and the Gauß-Legendre procedure, but is only available for KrC, ZBL and Molière. Evaluation of the accuracy [14] of the integration procedure Magic shows a maximum relative error of the scattering angle in the center-of-mass system of about 1 % nearly independent of the relative impact parameter (impact parameter/screening length). For the Gauß-Mehler procedure the corresponding error is increasing with an increasing relative impact parameter and depends on the number of pivots [14]. According to Robinson [15], the Gauss-Mehler method is generally more accurate than the Gauss-Legendre method in evaluating the scattering angle integral, but less accurate for the time integral.

3. Structure of the code

The code SDTrimSP treats the bombardment of incident ions on different target structures. Besides mono-atomic targets, layer structures, fixed and variable composition target structures are allowed. The kind of projectiles and/or target atoms is not limited. Both incident ions and recoil atoms are treated as series of subsequent collisions. There are two general cases in the code:

- static case: the target composition is fixed during the whole simulation
- dynamic case: modifications of the target caused by the ion bombardment are taken into account; in this case the target is updated at regular intervals, i.e. after a certain number NR$\geq$ 1 of projectiles and corresponding showers. NR has to be specified as a parameter in the input file.

The atoms are distinguished in projectiles (incident atoms) and recoils (target atoms). For each traced atom important physical quantities, as energy, spatial coordinates, direction of motion, are recorded along its path using general data structures. Moreover, the path length and the number of collisions are stored for the projectiles, while for the recoils the collision number in which they are generated is stored (generation). Besides the information about the single projectile there are also quantities integrated over all projectiles to save memory. For projectiles, the inelastic (electronic) energy loss and the total elastic and the elastic loss larger than the displacement energy are stored. Other values derived from these basic quantities can be determined, if of interest.

The structure of the program is depicted in Fig. 2. In the projectile loop, groups of NR projectiles are followed from collision to collision. The recoils generated along the projectile trajectories of the NR incident ions are collected and treated in a separate loop, the recoil loop. After finishing the calculation of the NR projectiles and generated recoils the target is updated in the case of the dynamic mode. In the static mode no target update is necessary, and it can be continued with the next group of projectiles until the total number NH of projectiles (number of histories) is reached. Finally, the output section is entered.

In the input file the target and incident particles are specified. A flag determines the static or dynamic mode. In the dynamic case the total fluence for a calculation has to be given in units of 10^{16}atoms/cm^2. The energy of the incident particles, the angles of incidence, the interaction potential and the inelastic energy loss model have to be chosen. The energy and angle of incidence of the projectile can be chosen fixed or by a given distribution. The input file is organized as a F90 namelist file and described in detail in the documentation delivered with the program package.

The output was designed in a very flexible manner allowing to store all important values of individual particles and offering at the same time the possibility to limit the output in order to save memory and computing time. By conditioning the different output sections in the code the user can switch on or off the different sections with corresponding flags and variables in the input file. Moreover, the output is structured in such a way that the user can insert own output sections in an obvious manner.

The general, obligatory output gives the reflection and sputtering coefficients, atomic fractions and densities as a function of depth, and the yield versus the generation. In the dynamic case the change of target thickness and atomic fractions and densities as a function of fluence is given. This minimal output has a size of some kBytes only. Optional output concerns trajectory information (evolution of spatial coordinates, directions of motion, energy, time), particle information (energy, number of collisions, path length, starting point and final coordinate), matrices (absolute frequency distributions of reflected, transmitted and sputtered particles in discrete levels of energy and exit angles). Note, that the amount of output can increase rapidly to hundreds of MBytes for the trajectory and particle output, especially when the incident energy is high. Especially for problems with a large number of reflected, transmitted and sputtered particles, the usage of matrices output is advantageous as it helps to save memory. There are several post processing programs concerning the matrix output and the visualization of calculated data by means of IDL.

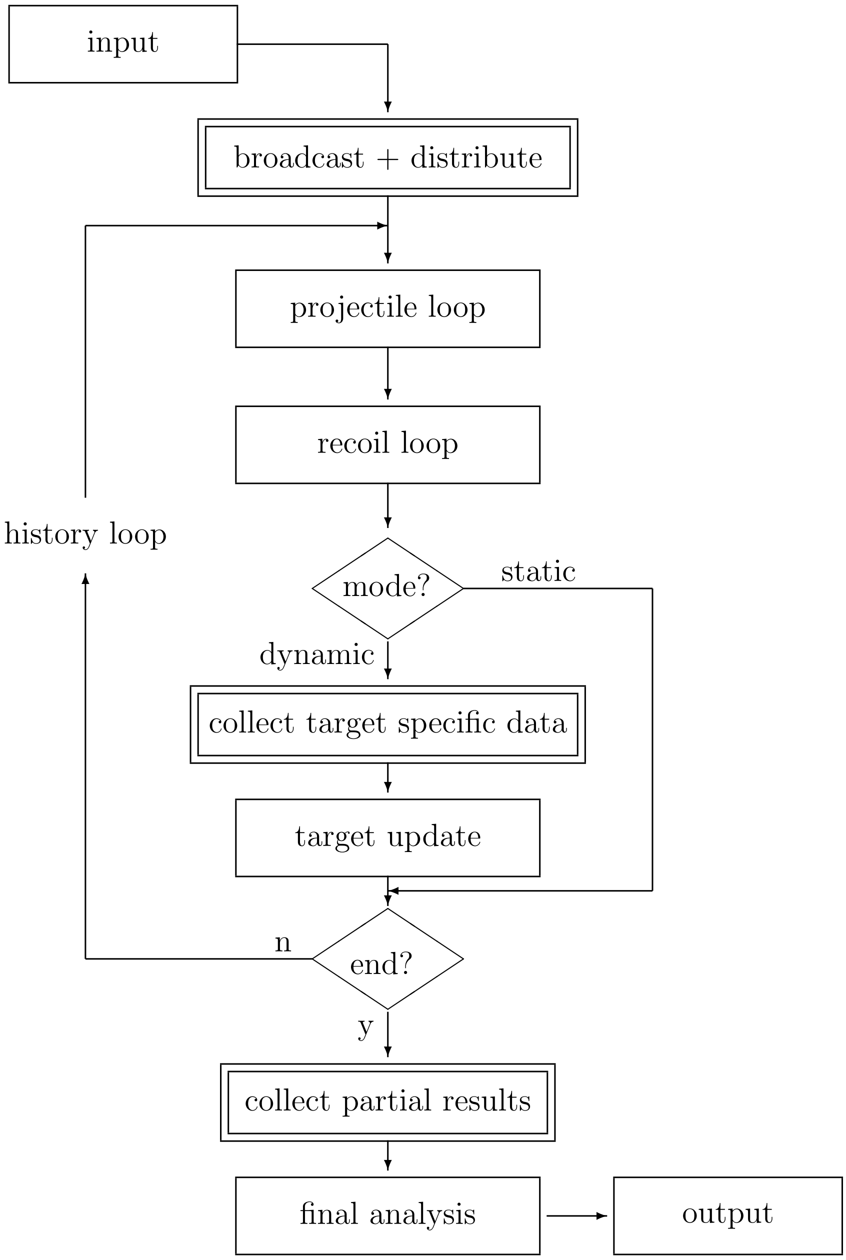

Figure 2: Main block flow chart of the program SDTrimSP. The double blocks indicate additional work necessary in the parallel mode. These parts are skipped in the sequential mode. Target updates include diffusion and chemical reactions.

4. Implementation

The program SDTrimSP is implemented in Fortran 90. The work flow depicted in Fig. 2 is transferred into a modular structure of the code. The characteristic quantities belonging to larger data units like trajectories, particle states and other data blocks are combined in F90 user-defined structures to make data handling easier.
The code is drawn up to work in different modes and on different architectures. Basically, there are two modes:

1. the sequential mode for execution on any sequential architecture with a F90 and a C compiler,

2. the parallel mode for execution on any distributed-memory parallel architecture with a F90 and a C compiler and the MPI communication library available. In this version, the NR particle showers are distributed over the processors.

Both modes are included in the same source code, the different modes are distinguished by use of preprocessor directives. The user selects the mode at compile time by choosing the respective macro in the Makefile.
In the sequential mode the course of the program is as shown in blocks with simple boxes of Fig. 2. In the parallel mode some additional work as specified blocks with double boxes in Fig. 2 is necessary to distribute data structures and computational load and to summarize the results. The parallel algorithm works as follows: The target data is replicated on all processors (broadcast in Fig. 2), while the NR incident ions between two target updates are distributed over np processors (NR $\rightarrow$ NR/np). Furthermore, each processor has to be provided with an appropriate seed for the random number generator (see below). The ions and corresponding showers are simulated independently on the processors, each processor using a dedicated sequence of random numbers. The effects caused by the particles are recorded in processor private variables. This concerns target data, ion specific data and recoil specific data. In the dynamical case, the target data has to be summed up over the processors and made known to each processor in order to perform the target update. The target update is carried out quasi-sequentially on all processors and as a result each processor has a replicate of the new target data and can continue with the next group of ions and so on. In the static case the target update and the global sums are not necessary for the calculation and the summation of the target data is postponed to the end of the program where in any case all particle information gathered locally on the processors has to be collected and printed out. That means the static program is embarrassingly parallel with nearly no communication, while the amount of communication in the dynamic case can be considerable. Depending on the application the computing time may be rather long, therefore restart files can be written at regular time intervals.
The communication is based on the Message Passing Interface (MPI). By this the algorithm is portable between different parallel architectures. Special care has been taken for the generation of random numbers. By choosing the linear congruential random number

generator from Cray's Scientific Library we have a true parallel random number generator having the advantage that the sequence of 2^{32} values can be divided into chunks of equal size so that each processor has its own sequence which is not correlated to the sequences of all other processors. To facilitate debugging and providing reproducible results there is also the possibility to associate each particle with its own, determined seed so that the result of a computation with a certain number of incident particles is always the same irrespective of how many processors are involved. One must, however, be aware that this method of using reproducible random numbers does not yield reliable results in the sense of good statistics. Again, the mode of random number generation is controlled via preprocessor directives in the code and corresponding macros in the Makefile.

The calculation steps in the program SDTrimSP are determined by NR and NH. NR is number of projectiles between target updates, NH is the number of histories. In the dynamic case the target is relaxed after each history step.

Note that the structure of the whole package SDTrimSP is designed in such a way that the same source code, Makefile and run-time commands are used for all modes and architectures and distinctions are made via preprocessor and environment variables (e. g. OSTYPE). The object code is kept in different directories for the different architectures to facilitate the simultaneous usage of different architectures. A detailed description of the code with a list of all input and output variables and a description of all subroutines with references to the corresponding literature is provided. The code is available for free for non-commercial use. (contact mail-address: SDTrimSP@ipp.mpg.de or see the webside: www.ipp.mpg.de/˜stel/SDTrimSP.html).

5. Performance

The program was tested on several sequential and parallel architectures, as e. g. IBM SP machines, IBM Power4 and Power5 systems, Cray T3E, NEC SX5, and Linux clusters with AMD or Intel processors, and is running in production mode for several years now with great success. For large, time-consuming applications it is advisable to use the parallel version of the code. In this case, the choice of the parameter NR, which is the number of incident particles and corresponding showers between two target updates, is decisive to have good performance, while in the sequential version, this parameter is of no relevance. The reason is that NR is a quantity closely related to the granularity of the parallel algorithm, as each processor has to treat NR/np incident particles together with their recoils, where np is the number of processors. That means, NR must not be less than np, and the larger NR, the better the efficiency of the parallel program, as the ratio between communication and computation decreases. On the other hand, NR has also a physical meaning, as it determines somehow the frequency of target updates. Therefore an investigation of the influence of NR and NH on the accuracy of the results has been carried out.

5.1. Influence of NR and NH on the accuracy of the results

A physical interpretation of NR and NH is that a larger number NR improves the statistical relevance of the target update, while a larger number of histories NH means a smaller fluence step (because the fluence step is the total fluence divided by NH) and by this inproves the overall statistics. In the static case, the statistics depends only on the product of NR and NH, and NR has no physical meaning.

With the following example of a dynamic case it is shown that the accuracy of the results depends merely on the product of NR and NH over a certain range of values for NR. The chosen example is a 1 keV bombardment of Fe on TaC with a fluence of 10^{17} atoms/cm^2. The results for different values of NR with $\mathrm{NR} \cdot \mathrm{NH} = constant$ are shown in Table 1 and Figs. 3 and 4 at the example of typical quantities. It shows that the plots and values differ only to the same extent as they would differ when using another seed for the random numbers. That means, the number NR may be increased to achieve a better parallel efficiency, while decreasing at the same time the number of histories. For statistical reasons it makes no sense to choose very small values of NH in the dynamic case. The fluence step (total fluence/NH) should of the order of 0.01 (10^{14} atoms/cm^2) to ensure that the target composition change is small in a fluence step.

NR	NH	change of thickness	qu(Fe)	qu(Ta)	qu(C)
1	1000000	4.47 nm	0.559	0.302	0.139
10	100000	4.51 nm	0.557	0.303	0.140
100	10000	4.50 nm	0.557	0.304	0.139
1000	1000	4.49 nm	0.558	0.303	0.139

Table 1: Change of thickness and atomic fraction (qu) of the surface composition with different numbers of NR and NH for the example of Fe → TaC ($\mathrm{NR} \cdot \mathrm{NH} = constant$)

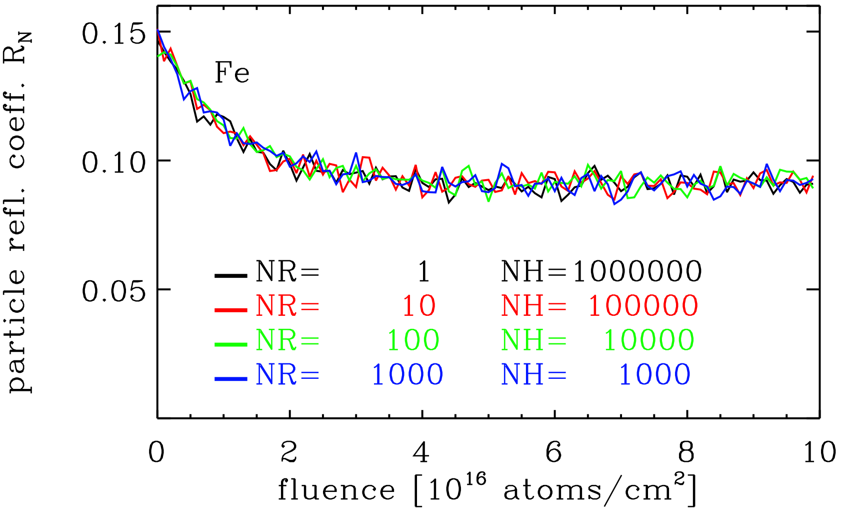

Figure 3: Particle reflection coefficient for different numbers of NR and NH ($NR \cdot NH = constant$) in the case of 1 keV Fe atoms impinging at normal incidence onto a TaC target

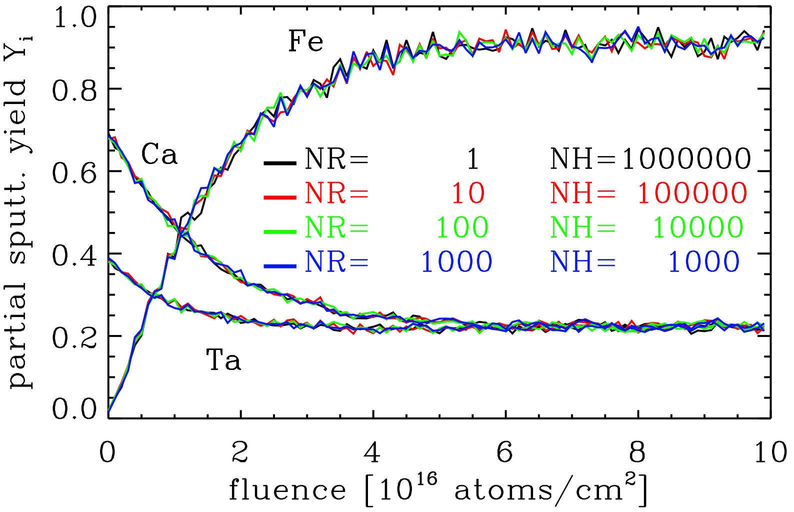

Figure 4: Partial sputtering yield for different numbers of NR and NH ($NR \cdot NH = constant$) in the case of 1 keV Fe atoms impinging at normal incidence onto a TaC target

5.2. Parallel efficiency

As already pointed out the performance of the program depends strongly on the mode and on the choice of NR on the one hand, but on the other hand on the characteristics of the used architecture. When working on a single-processor system, i. e. using the sequential version, it is mainly the clock rate of the processor which determines the computing time, irrespective of whether using the static or dynamic mode. The choice of NR is not of great importance. In the dynamic case, however, the code is rather communication-intensive, and the performance depends heavily on the choice of NR.

These dependencies are demonstrated at the example of 1 keV Fe bombardment on TaC at normal incidence with a fluence of 10^{16} atoms/cm^2. The benchmark has been carried out on two different parallel architectures, an IBM 1.3-GHz-Power4 (Regatta) system and a Linux cluster with Intel 2.8-GHz processors. The Regatta system is provided with a fast communication network with Federation switch, while the nodes of the Linux cluster are connected via Gigabit Ethernet. For these benchmarks the option of minimum output was used, the parameters NR and NH have been chosen as NR = 512 and NH = 20000 which allows to use up to 512 processors in the parallel mode.

The execution times and parallel efficiencies obtained for the static mode on the two architectures are shown in Table 2. As expected the parallel efficiency of the code is very good on both architectures, because the amount of communication is very low and consists mainly in broadcasting the data at the beginning of the calculation and summing up the partial results of the processors at the end of the calculation. This is also reflected in the corresponding speedup curves, cf. the solid lines in Fig. 5(b).

Table 3 and the dashed lines in Fig. 5(b) show the corresponding behavior for the dynamic case. There is a clear difference in the performance for the two architectures. The parallel efficiency obtained with the IBM Regatta is very good up to 64 processors. This is due to the fast communication achieved by the strong Federation switch of the Regatta system and the MPI implementation on top of the shared memory architecture of the Regatta. In contrast, the parallel efficiency on the Linux cluster is not that good. This is due to the fact that the communication network of the Linux cluster is rather slow compared to the processor performance and cannot cope with the amount of communication. The speedup curves demonstrate the somewhat poorer scaling and show that the Linux cluster is not specially suited for parallel calculations with more than 16 processors in the dynamic mode. Up to 16 processors, however, the performance of the Linux cluster is quite satisfactory, at least for the chosen value of NR = 512. It should be noted that the single-processor performance of the Linux cluster is much better than that of the IBM Regatta.

To conclude, the mode of the calculation, the choice of different parameters and the characteristics of the parallel architecture determine the efficiency of the calculation. To improve the performance in the dynamic mode it is advisable to reduce the communication by using a small number of NH and a large number of NR.

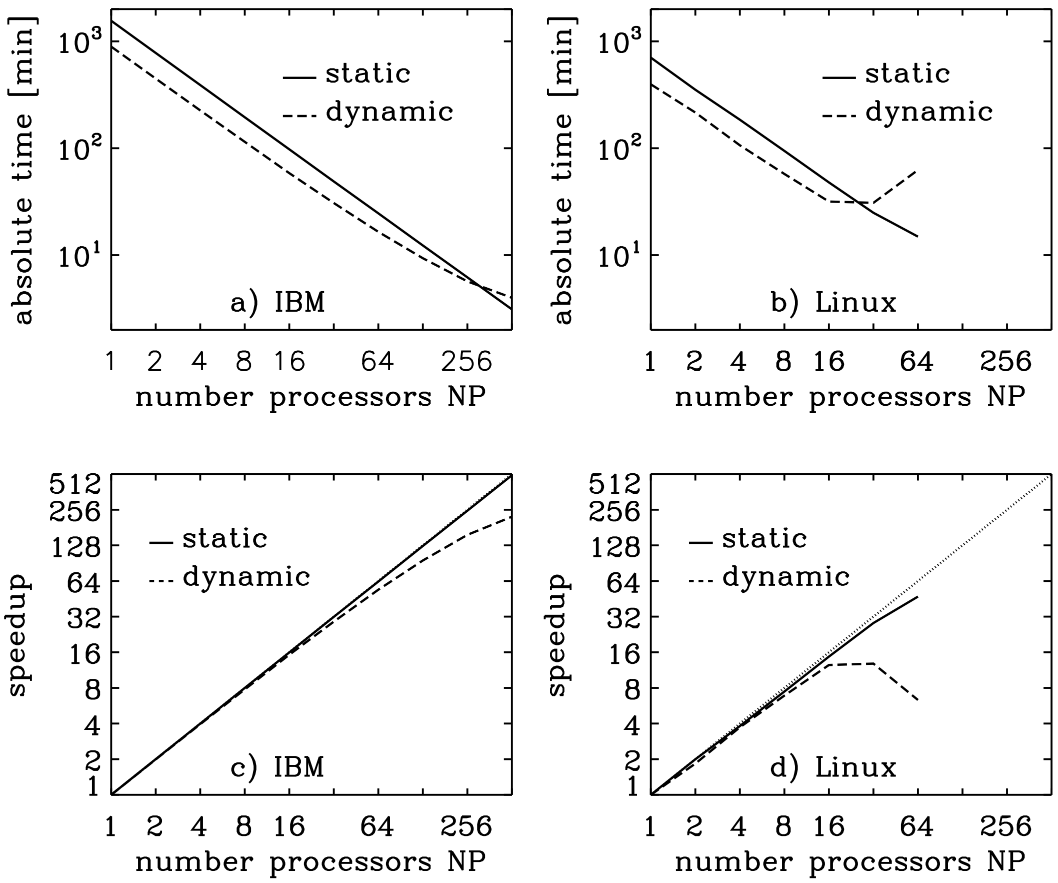

Figure 5: a, b) Absolute time and c, d) speedup of the static and dynamic cases in dependence of the number of processors for the IBM Regatta (IBM) and an Intel Linux cluster (Linux)

	IBM Regatta 1.3 GHz		Linux cluster 2.8 GHz (Intel)	
	execution time	parallel efficiency	execution time	parallel efficiency
np	[min]		[min]	
1	1557.38	1.000	703.84	1.000
2	778.61	1.000	352.47	0.998
4	389.96	0.998	184.78	0.952
8	194.81	0.999	94.40	0.931
16	97.65	0.996	47.86	0.919
32	48.87	0.995	24.90	0.883
64	24.86	0.989	14.89	0.738
128	12.31	0.988		
256	6.20	0.981		
512	3.11	0.978		

Table 2: Execution time and parallel efficiency of the static version of SDTrimSP on the IBM Regatta and on an Intel Linux cluster for the example of 1 keV Fe atoms impinging at normal incidence onto TaC with NR = 512, NH = 20000 (10240000 particles)

	IBM Regatta 1.3 GHz		Linux cluster 2.8 GHz (Intel)	
	execution time	parallel efficiency	execution time	parallel efficiency
np	[min]		[min]	
1	891.90	1.000	395.90	1.000
2	448.17	0.995	215.72	0.917
4	225.84	0.987	106.48	0.929
8	114.77	0.971	57.46	0.861
16	58.47	0.953	31.69	0.780
32	30.72	0.907	30.85	0.401
64	16.51	0.844	62.58	0.098
128	9.35	0.745		
256	5.68	0.613		
512	3.99	0.436		

Table 3: Execution time and parallel efficiency of the dynamic version of SDTrimSP on the IBM Regatta and on an Intel Linux cluster for the example of 1 keV Fe atoms impinging at normal incidence onto TaC with NR = 512, NH = 20000 particles

6. Special applications

6.1. Static mode

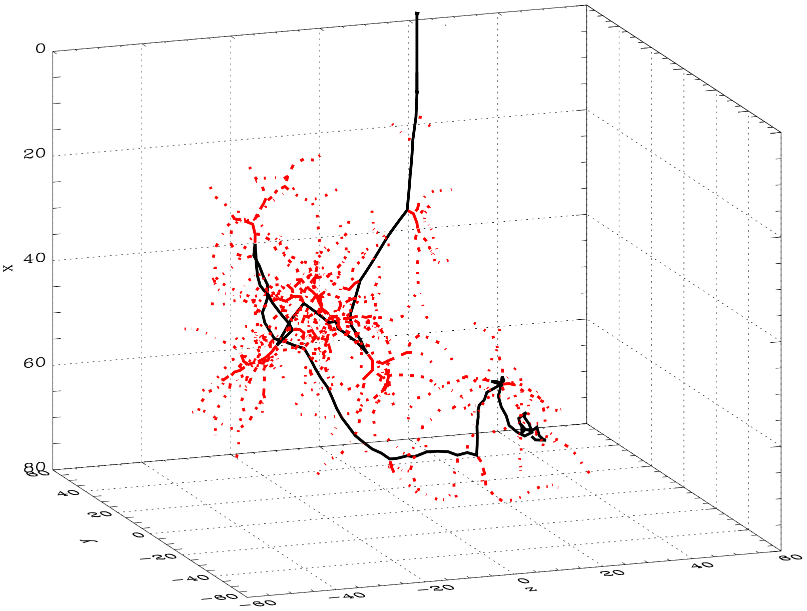

Figure 6: Trajectory of a 2 keV He atom penetrating a Ni target at normal incidence. The color indicates the projectile (black) and the recoils (red).

Trajectories A typical trajectory of a 2 keV He atom in a mono-atomic Ni target is shown, see Fig. 6. The path of the incident He atom, the projectile, is black and the paths of all recoils are red.

The decreasing energy of the atom along its path through the solid is indicated by the color, Fig. 7(a). The atom is stopped if its energy is smaller than the cutoff energy which is chosen to be 1.0 eV. In Fig. 7(b) the same trajectory as in the preceding figure is shown together with the generated Ni recoils. The recoils of the first generation are indicated in red, the recoils of the second generation in blue.

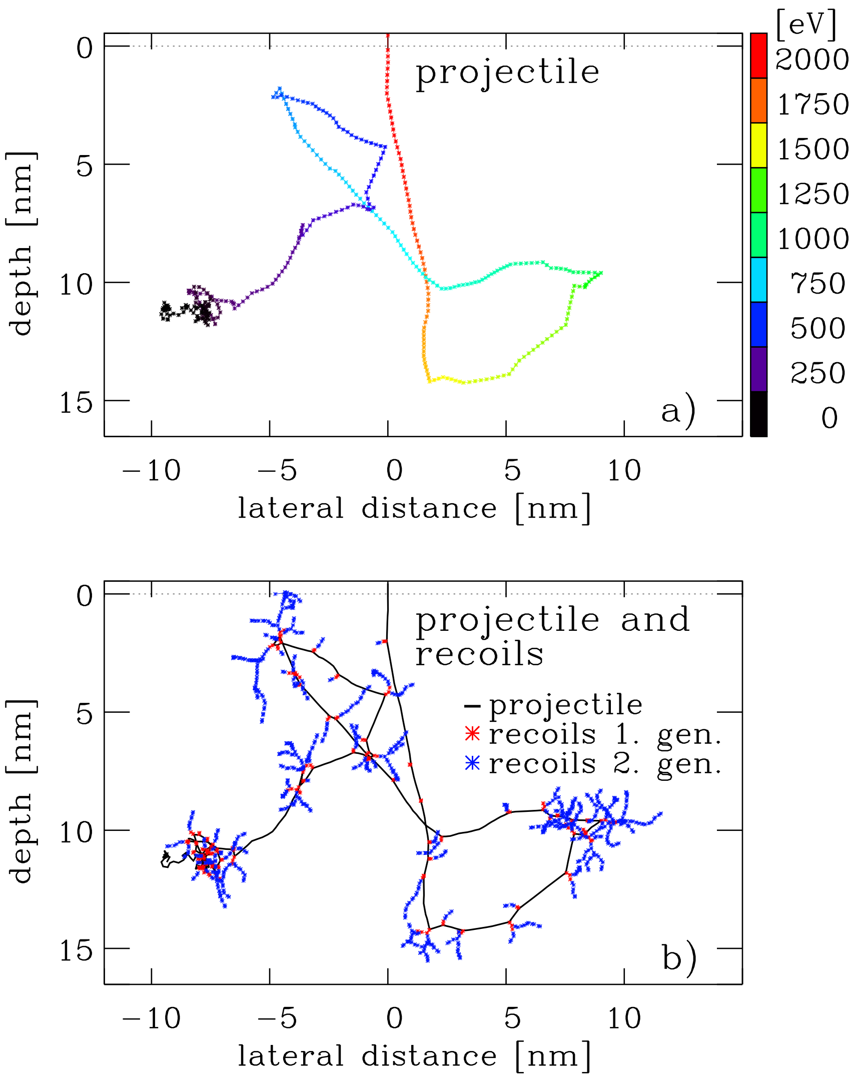

Figure 7: Trajectory of a 2 keV He atom penetrating a Ni target at normal incidence. a) The color indicates the decreasing energy of the He atom along its trajectory. b) In addition to the He trajectory also the generated recoils are shown. The color indicates the recoils of first generation (red) and the recoils of the second generation (blue).

Angular distribution of scattered and sputtered atoms

Here, we consider the bombardment of an Ni target with 1 keV Ar at 60°. The contour plots for the angular distributions of the reflection coefficient and of the yield per solid angle are shown in Fig. 8. The largest coefficient for the backscattered particles is reached in the forward direction at an polar angle of about 75°, whereas for the backsputtered atoms a high-intensity ridge appears in the forward direction up to an azimuthal angle of 40°. The highest yield per solid angle occurs in the forward direction at about 30°. The lowest coefficients are in the backward direction for both kinds of particles.

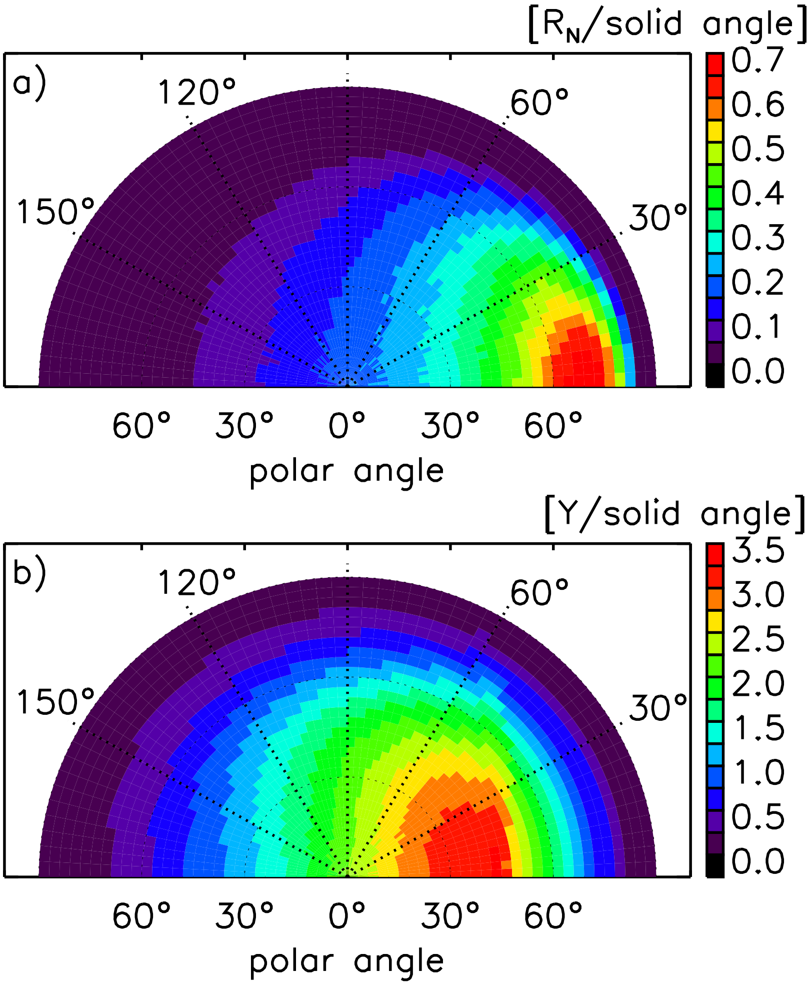

Figure 8: Contour plot of the angular distribution of the coefficient per solid angle of a) backscattered Ar atoms and b) sputtered Ni atoms. A Ni target is bombarded with 10^8 1 keV Ar atoms at 60°. The intensity is indicated by the color.

6.2. Dynamic mode

Backscattering coefficient, sputtering yield and atomic fraction: Backscattering coefficient, sputtering yield and atomic fraction Fig. 9 shows the fluence dependence of the particle backscattering coefficient, R_N, and the partial sputtering yields, Y_i, for the bombardment of the compound target WC with 10 keV Ni at normal incidence. At a fluence of about 10^{17} atoms/cm^2 R_N and Y_i become constant, which means that steady state or equilibrium is reached. The backscattering coefficient of Ni is decreasing with increasing fluence because some of the heavy W atoms are replaced by the lighter Ni atoms. This can be seen in Fig. 10, where the atomic fractions of the three species are shown versus depth at different fluences. The partial yield of Ni is increasing from zero (pure WC target) to a constant value. At steady state the amount of Ni in the target is not changing any more with fluence which means that $R_N + Y_{\mathrm{Ni}}$ must be unity. It can also be noted from Fig. 10, that the depletion of C in the target is larger than that of W. It is a well-known fact, that in many cases the lighter element in a multi-component target is preferentially sputtered. It should be remembered that diffusion and segregation effects are not included in the calculations.

Dynamic changes of the target composition: Another interesting case is the bombardment of a target consisting of light atoms by heavy ions, in this case the bombardment of C by W at normal incidence. At the beginning of the bombardment, the target swells (positive value of surface position). This is a result of the deposition of W into the carbon target, which is larger than the sputtering of C. The composition of the target is changed particularly after a fluence of $5 \cdot 10^{16}$ atoms/cm^2. Therefore, the sputtering and reflection of W starts and the target shrinks (negative value of surface position), see Fig. 11(a). The values of backscattering and sputtering change quasi-periodically according to the composition of the target, see Fig. 11(d). The peak of the partial yield of W (Y_{Si}) appears when the peak of the W implantation profile reaches the surface; the selfsputtering of W is much larger than the sputtering of C by W. The occurrence of further peaks is caused by the generation of further implantation profiles of W until they die out. After a fluence of $30 \cdot 10^{16}$ atoms/cm^2 a static state or equilibrium is reached and the coefficients R_N and Y_i get constant, see Fig. 11(b,c). The calculated results for atomic fraction of W show good agreement with experimental data [27], see Fig. 12.

Target composition: The program allows also layered target structures. As an example a target with several Si and Ta layers on Si is chosen, which is bombarded at normal incidence with 3 keV Ar.The oscillatory behavior of R_N and Y_i originates from the layered structure. For R_N the reason for the maxima is the higher reflection coefficient of Ar from Ta compared to that from Si due to the different mass ratio of target atom to incident ion. The peak of Y_{Si} at a fluence of about $5 \cdot 10^{17}$ atoms/cm^2 originates from the higher backscattering of Ar from the underlying Ta. Fig. 13 shows the broadening of the depth profile, the atomic mixing and the recoil implantation in the target. Again, in this example the lighter target element, Si, is preferentially sputtered. In this run, the implantation of Ar into the target is neglected.

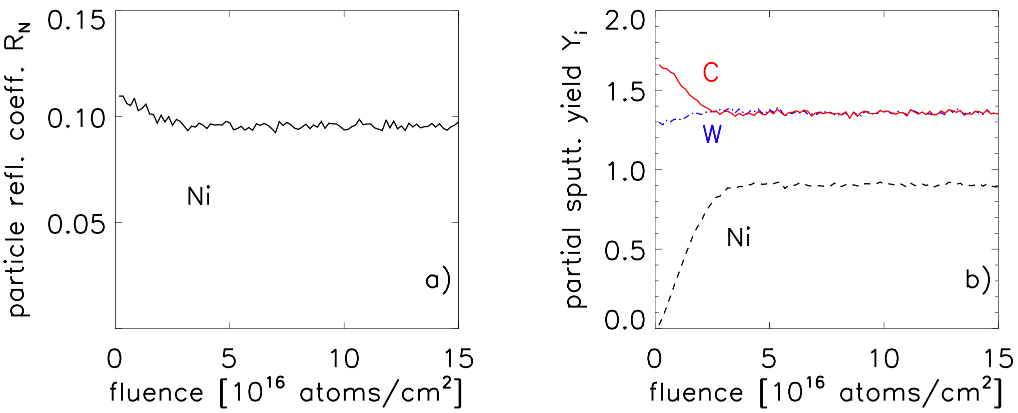

Figure 9: Fluence dependence of a) the particle reflection coefficient, R_N, and b) the partial sputtering yields, Y_i, by Ni on WC at normal incidence. A WC target is bombarded with 10 keV Ni.

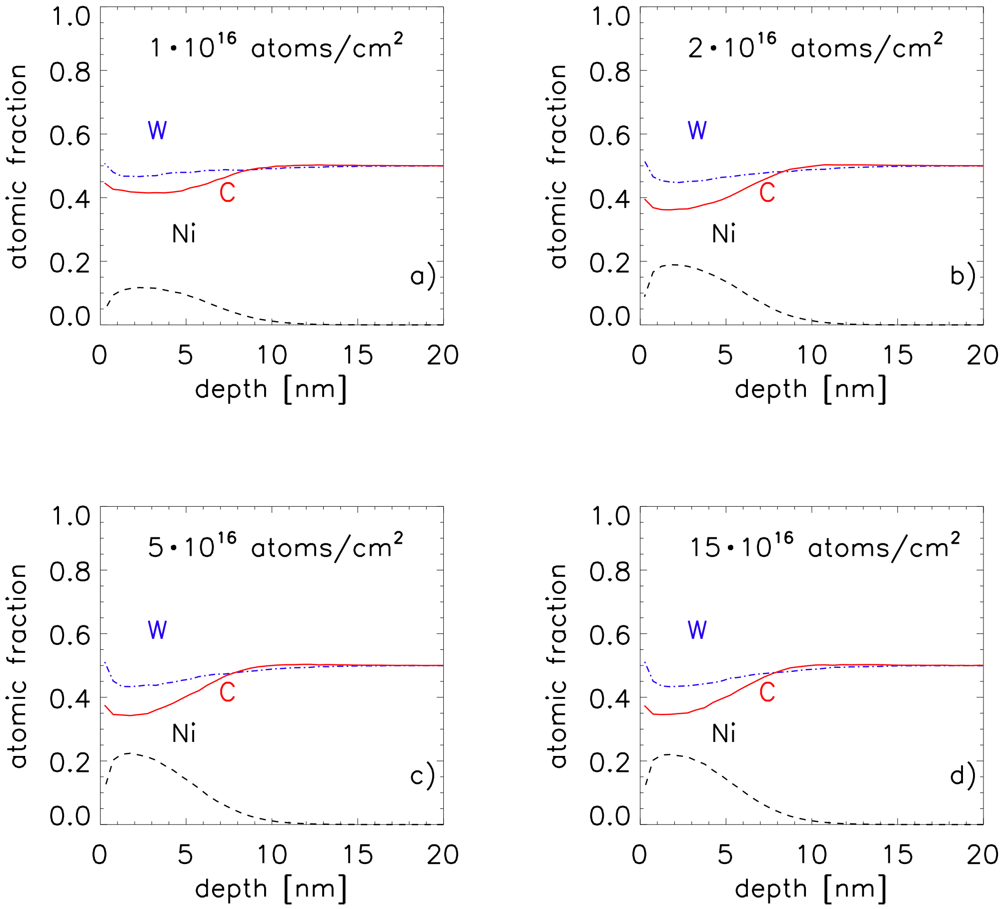

Figure 10: Atomic fraction of C, Ni and W versus depth dependent on fluence. A WC target is bombarded with 10 keV Ni at normal incidence.

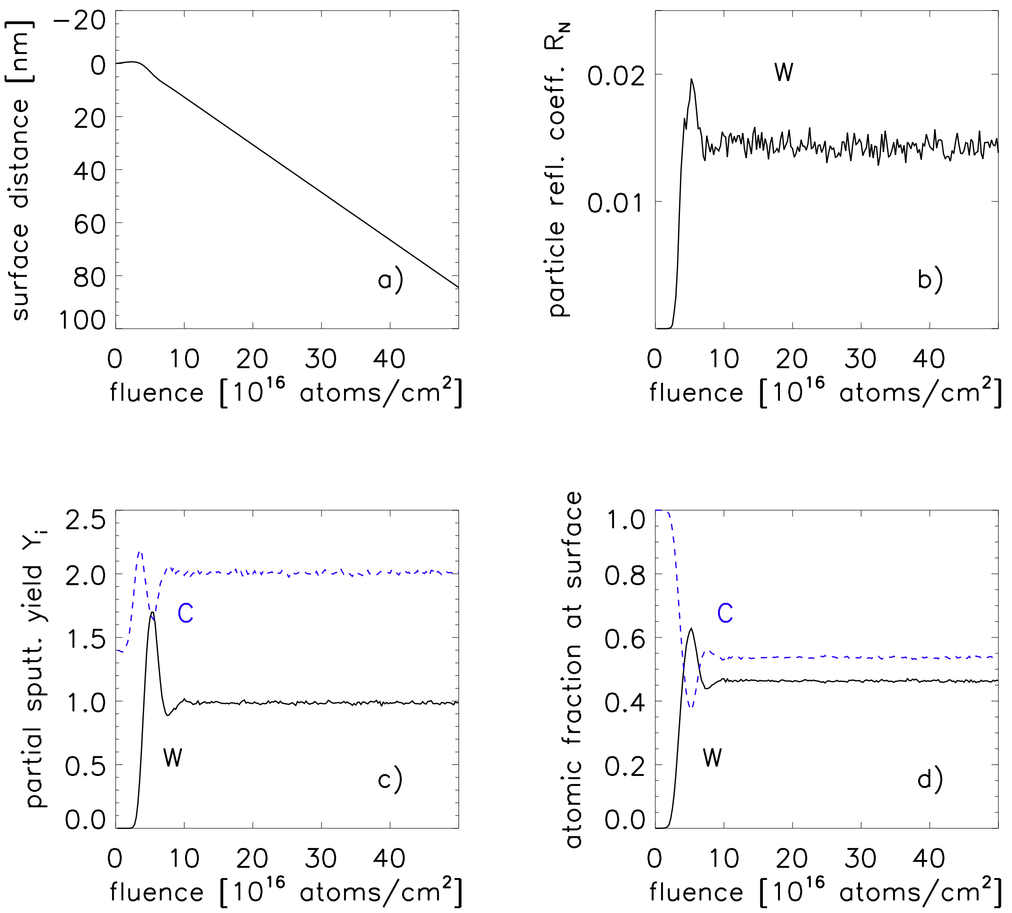

Figure 11: a) Surface distance, b) Particle reflection coefficient, R_N, of W, c) partial sputtering yields, Y_i, of C and W and d) atomic fractions of W and C for the bombardment of a C target at normal incidence with 5 keV W atoms.

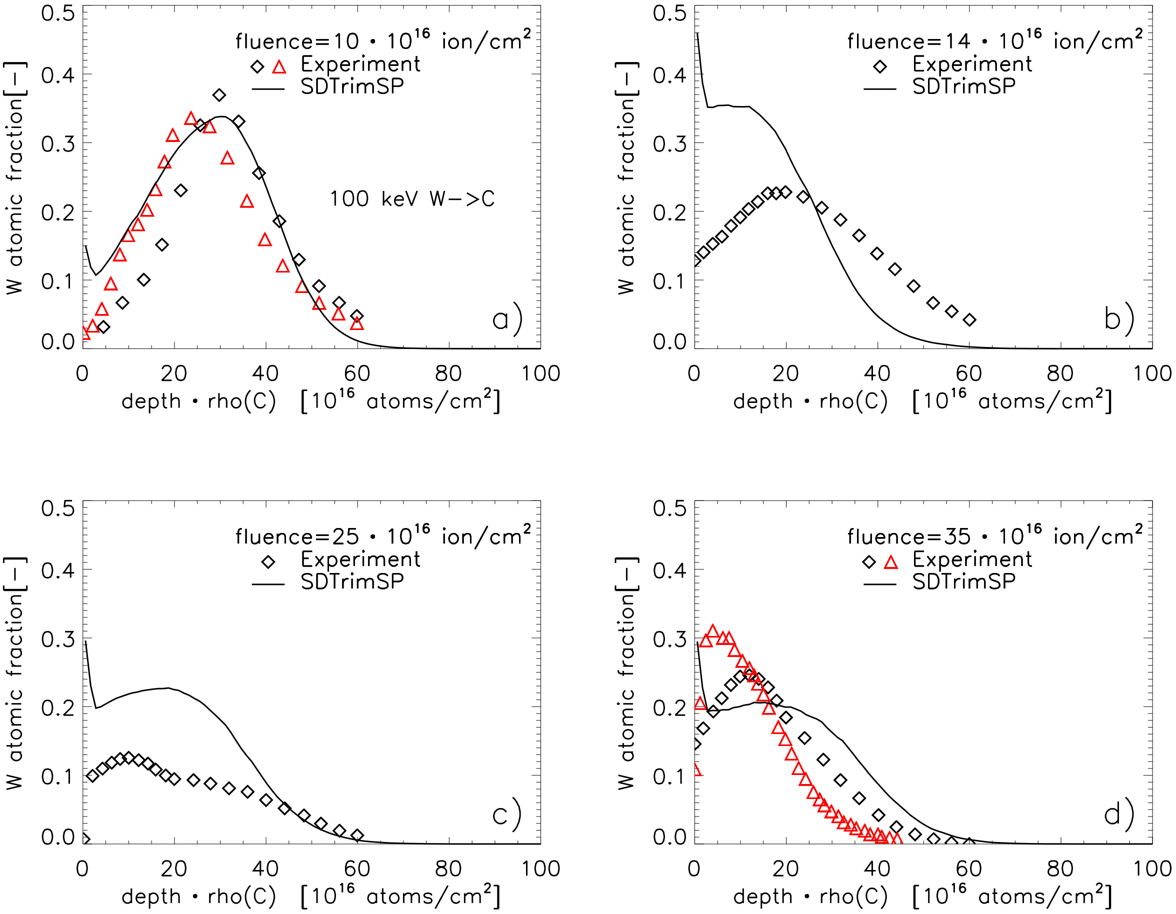

Figure 12: Comparison of the calculated atomic fractions of W with measurements [27] for four different fluences for the bombardment of a C target at normal incidence with 100 keV W atoms.

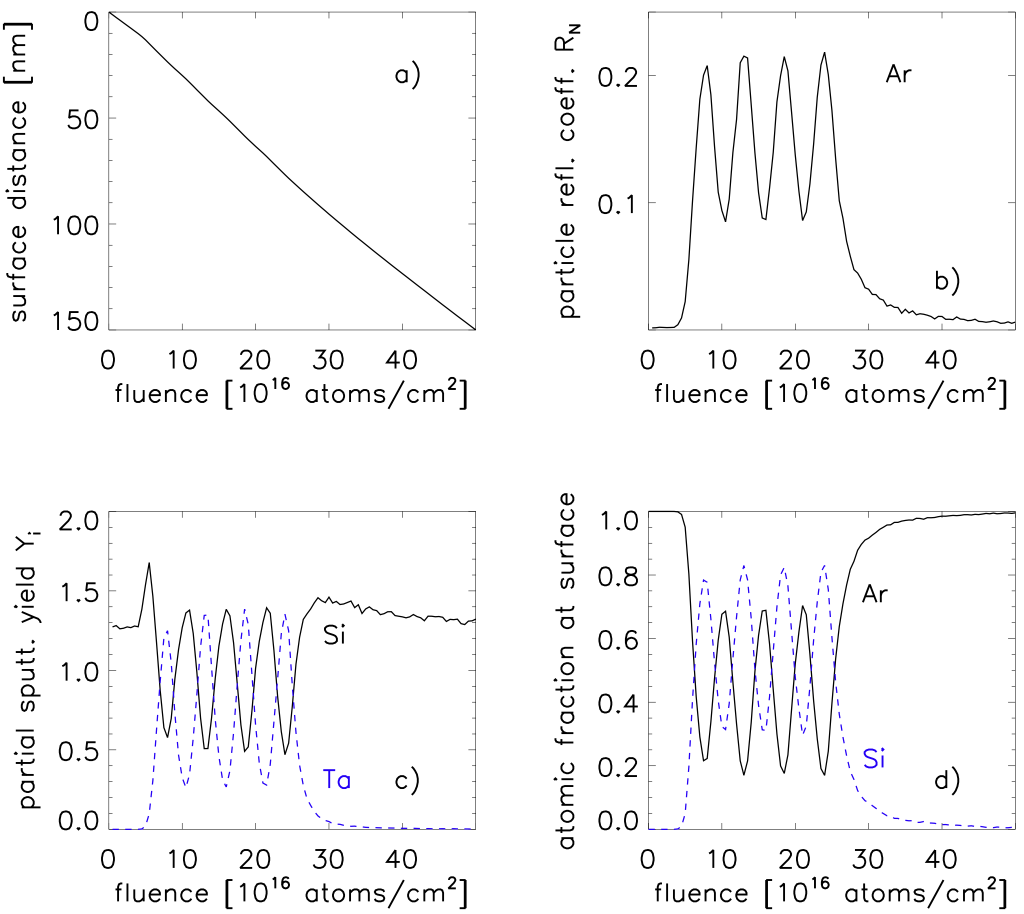

Figure 13: A Si(20 nm)[Ta(7.5 nm)Si(10.5 nm)]$_3$Ta(7.5 nm)Si target is bombarded by 3 keV Ar at normal incidence. a) Initial target composition, b) fluence dependence of the particle reflection coefficient, R_N, c) the partial sputtering yields, Y_i and d) the atomic fraction of Si and Ta at the surface (depth 0 - 0.5 nm).

7. Extensions in Version 6.00

7.1. Start of recoils

The standard calculations in version 5.00 were carried out without the use of displacement and bulk energy. The recoils start with the transfer-energy greater then the cutoff energy and stop the movement when their energy falls below some preset value or if they have left the target.

7.1.1. Bulk-binding-energy

In version 6.00 a recoil moves if the transfer-energy greater then the cutoff and bulk-binding-energy. The movement-energy of the recoil is reduced by the bulk-binding-energy. The bulk-binding-energy is a loss-energy. A problem is the surface-binding-energy. If a targetatom leaves the target both energies are subtracted. The result is a reduction of the sputtering yield. Fig. 14 shows that it is necessary to reduce the surface-binding-energy to get good results, if the bulk-binding-energy is not zero. The agreement with the measurements is only good, if the bulk energy is zero or the surface-binding-energy is reduced. For this example the surface-binding-model one (isbv=1) is used. The inelastic-model for Xe and Si is inel=3.

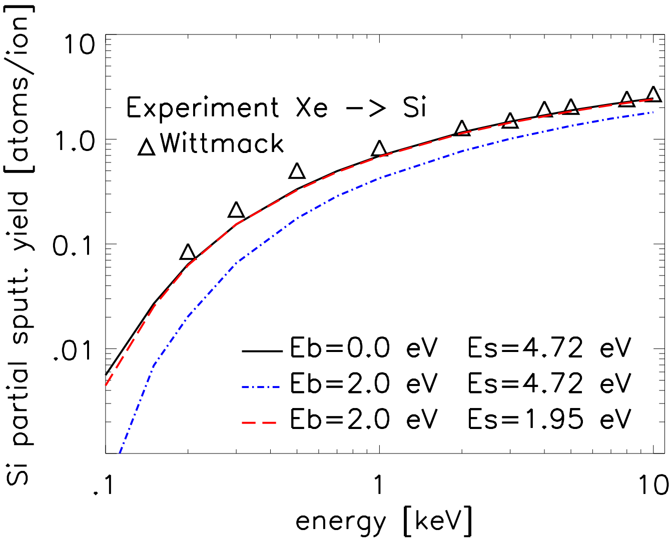

Figure 14: Calculated sputtered yield of Si for different bulk-binding-energies (Eb) and surface-binding-energies (Es) compare with experimental results from [31]

.

7.1.2. Displacement-energy (irc0)

The user can choose whether he wants to take into account the displacement-energy.
default: **irc0** =**-1**

If variable **irc0** $<$ **0** then the displacement-energy is not considered. The movement of all recoils with a transfer-energy greater than cutoff- and bulk-binding-energy are calculated, i.e. all recoils can leave their old places.

If **irc0**>**0** then the displacement-energy is considered (threshold). All movements of recoils with a transfer-energy larger than the cutoff- and bulk-binding-energy are calculated. But at the end the recoil may be reseted, i.e. the recoils with a start-energy less then the displacement-energy can not leave their old place. This is important for the consistent calculation of damages. If the transfer-energy is less then the displacement-energy no damage is produced.
This procedure (move also if the transfer-energy is less than the displacement-energy) is necessary because this threshold (displacement-energy) looses its validity near surface.

7.2. 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. The descriptions of the escape of noble gas atoms from targets in TRIM and SDTrimSP (versions 4.14) are inadequate. The effect of out-gasing in these programs was realized by the re-emission of gas atoms, namely the removal of atoms from the target without any transport of these atoms through the surface. In this case the knowledge of the maximum atomic fraction *qu_max* of the noble gas content in the solid is required for their removal. This introduction of an upper limit for the concentration of gas atoms, which must not be exceeded, does not describe the process itself and does not calculate profiles of gas atoms in the target, representing to the correct physical situation. Fig. 15 and 16 show the areal density and profiles without out-gasing (blue dashed-line) and the red dashed-lines show the calculated results with $qu\_max = 0.1051$. The problem is the determination of the value *qu_max*.
A first attempt to describe the out-gasing in the SDTrimSP using a gas transport model yields very good agreement of the fluence-dependent areal density values for energies larger than 10 keV, [21]. However, as shown in [22], the maximum of the atomic fraction as a function of the energy of the incident atoms and the stationary profiles of Xe are

not reproduced correctly. Also the physical explanation of a constant gas transport is difficult to provide.

To develop a model, that reproduces the steady-state profiles, areal density values and maxima of the atomic fraction, the following assumptions are used: The implanted gas atoms are not bounded in the target and cannot diffuse through a defect free lattice. The measured profiles after the bombardment of noble gas atoms on Si are the same as those produced during the bombardment. The implantation of gas atoms lead to a swelling of the target, so that a pressure arises, which is only gradually removed through relaxation. This pressure is responsible for local transport.

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 nonbounded 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.**

7.2.1. Damage-driven diffusion, DDF

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

$$J = -\frac{\partial(\eta(z) \cdot n(z))}{\partial z} \quad \text{, with} \tag{7.1}$$

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

$$\frac{\partial n}{\partial \phi} = -\frac{\partial(\eta(z) \cdot \frac{\partial n(z)}{\partial z})}{\partial z} \tag{7.2}$$

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 expresse as:

$$\eta(z) = \eta_0 \cdot P_{dam} \tag{7.3}$$

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:

$$P_{dam} = \frac{N_{dam}(z)}{max[N_{dam}]} \cdot qu(z) \tag{7.4}$$

The input variable of η_0 is **diff_koeff1**.
default: **diff_koeff1(:)=0.0**
An example for the variation of η or $P_{dam} = \eta(z)/\eta_0$ is displaces in Fig. 18 (red dashed line).

7.2.2. 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:

$$J = -K(z) \cdot n(z) \tag{7.5}$$

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:

$$\frac{\partial n}{\partial \phi} = \frac{\partial (K(z) \cdot n(z))}{\partial z} \tag{7.6}$$

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

$$\begin{aligned} K(z \leq z_{max}) &= K_0 & (7.7)\\ K(z > z_{max}) &= 0 & (7.8) \end{aligned}$$

The flux increases monotonically with the number of particles and the flux dependens on the form of the profile.
The input variable of K_0 is **diff_koeff2**.
default: **diff_koeff2(:)=0.0**
An example for the variation of K is provided in Fig. 18 (blue dot dashed line).

7.2.3. Results

The comparison between measured depth profiles and calculated profiles are shown in Fig. 17 and 18. The agreements of calculated areal density and maximum of atomic fraction with the measured values are very good, see Fig. 19 and 20.

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

$$\begin{aligned} \eta_0(Xe) &= 1.6 \cdot 10^6 \ \mathring{A}^4/ion \\ K_0(Xe) &= 70 \ \ \mathring{A}^3/ion \\ \eta_0(Ar) &= 1.6 \cdot 10^5 \ \mathring{A}^4/ion \\ K_0(Ar) &= 5 \ \ \mathring{A}^3/ion \end{aligned}$$

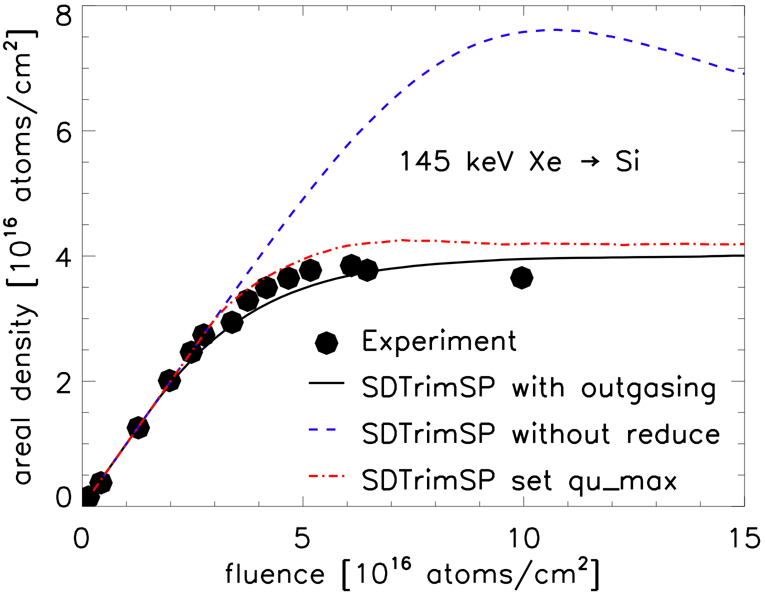

Figure 15: Comparison of experimental results [24] with calculated areal density 145 keV Xe on Si with and without transport of nonbounded particles and for $qu\_max = 0.1051$

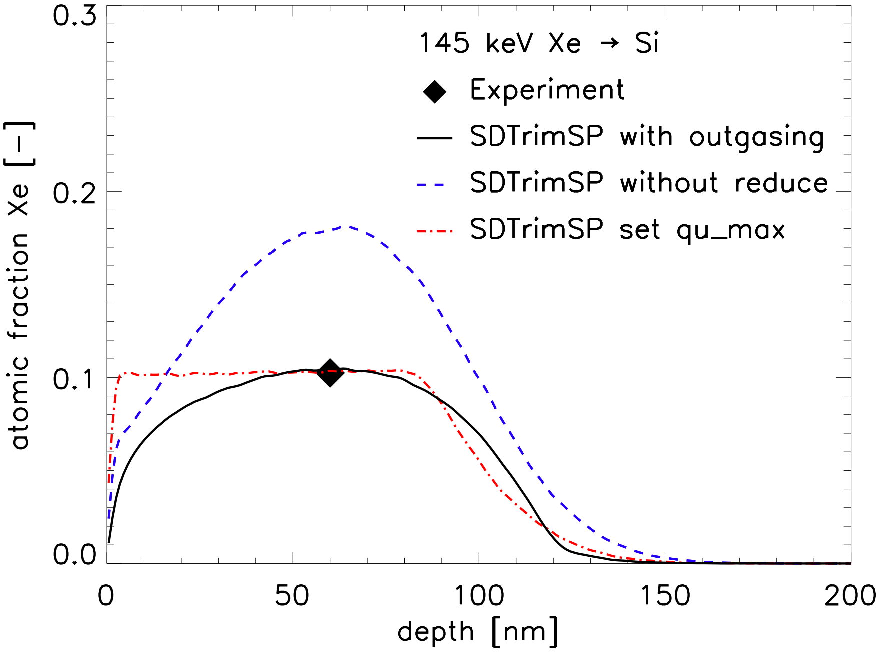

Figure 16: Comparison of the experimental maximum of atomic fraction [24] with calculated profiles for 145 keV Xe on Si with and without transport of nonbounded particles and for $qu\_max = 0.1051$

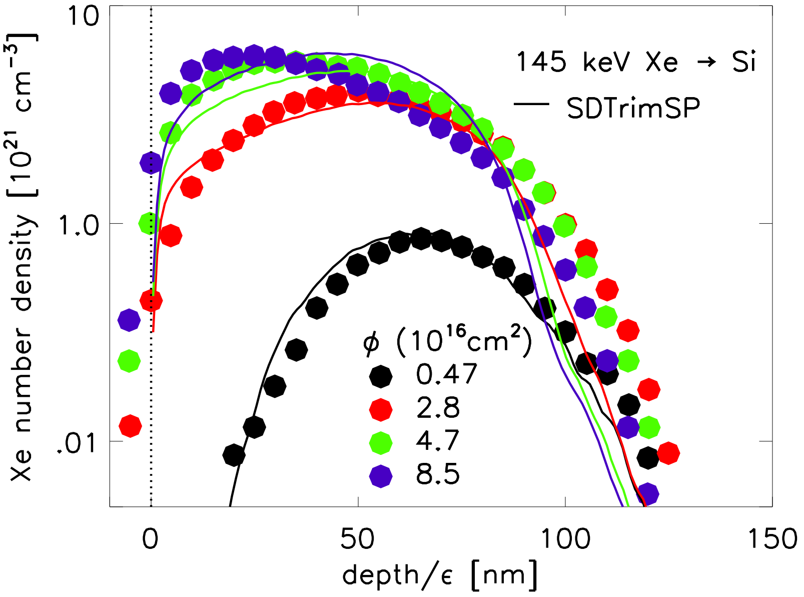

Figure 17: Comparison of experimental results [24] with calculated number density for 145 keV Xe on Si

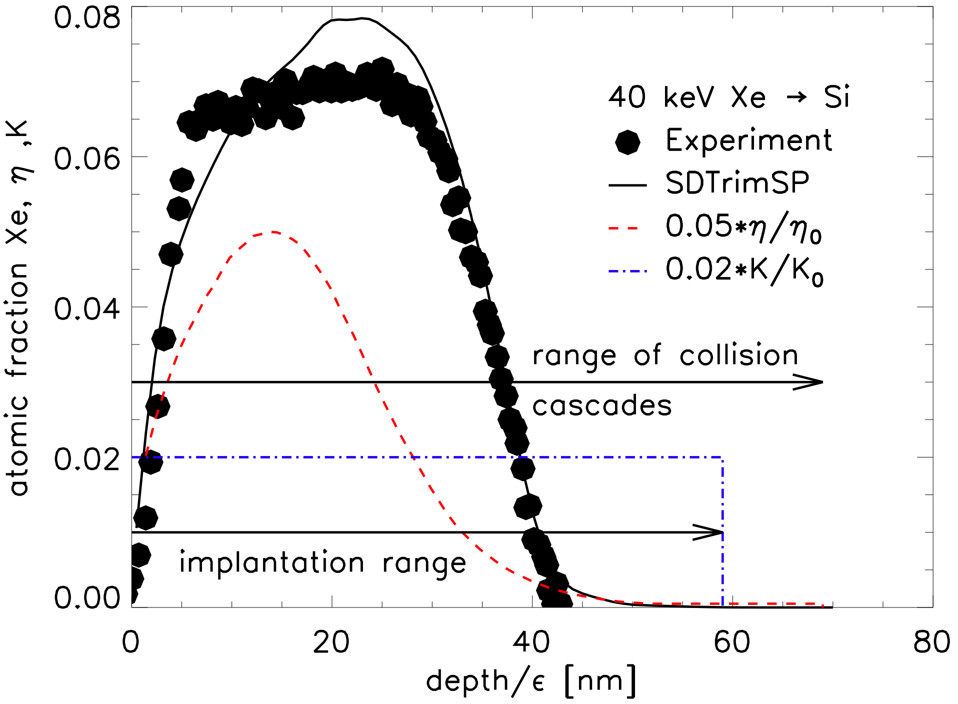

Figure 18: Compare of experimental results [23] with calculated atomic fraction for 40 keV Xe on Si with transport- and diffusions-coefficient

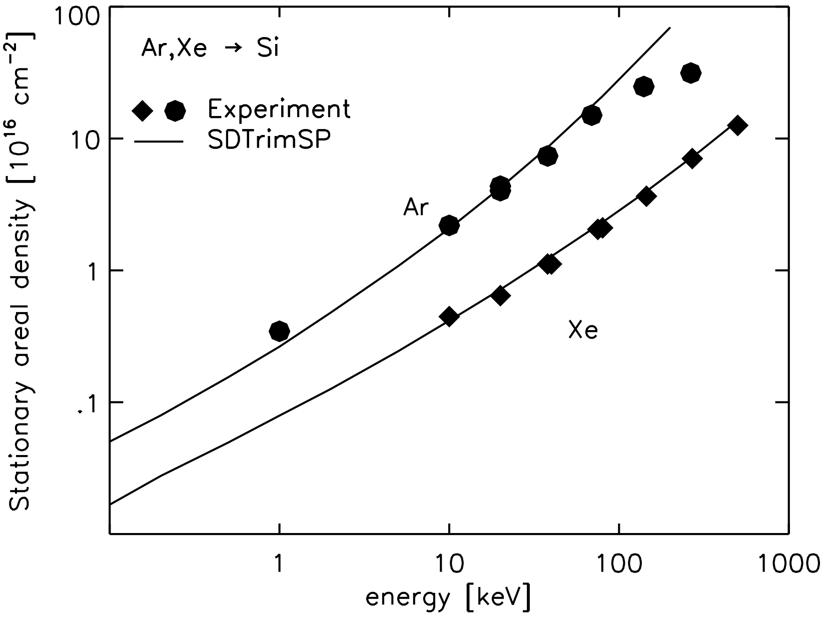

Figure 19: Calculated steady state areal density of implanted Xe versus the incident energy of Xe on a Si target at normal incidence compared with experimental data [22]

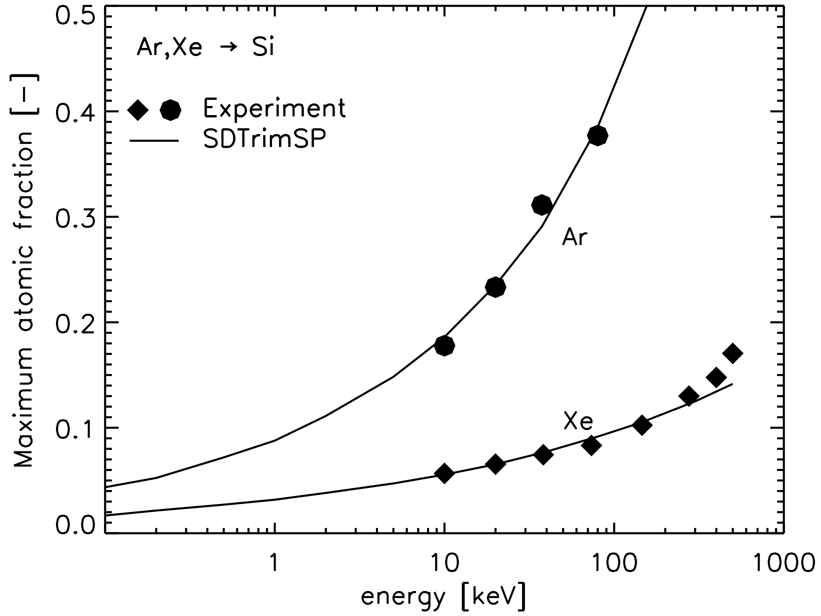

Figure 20: Calculated steady state maximum atomic fraction of implanted Xe versus the incident energy of Xe on a Si target at normal incidence compared with experimental data [22]

7.3. Chemical erosion of carbon

The total methane erosion yield (Y_{CH_4}) is calculated as a sum of a kinetic contribution Y_0 at $T_0 = 300K$ and a temperature-dependent thermal contribution Y_T, [35].

$$Y_{CH_4} = \; Y_0(E, T_0, sp^3, sp^x_{CH}) \; + Y_T(E, T, sp^3, sp^x_{CH}) \tag{7.9}$$

E is the energy of incident particles. sp^3 and sp^x_{CH} are carbon hybridization states.

The option is : **lchem_ch=.true.**,
default is: **lchem_ch=.false.**

Introduction of different carbon hybridization states
In the version of SDTrimSP without chemical erosion carbon atoms were simulated as a single species, C_g. For simulation of chemical erosion one needs to distinguish carbon atoms of different hybridization states as different species. Therefore, the chemistry module describes an a–C:H film in terms of three carbon species namely, sp^2 (carbon with double bonds), sp^3 (carbon with single bonds and not bonded to H atom) and sp^3_H (sp^3 carbon with a bonded H atom).

The chemical binding energy of C-C is 3.60 eV and of C=C 6.35 eV. The necessary energy to break the double-bond (C=C) to a single bond (C-C) is 2.75 eV. If the transfer-energy is greater than this value the carbon-hybrid sp^2 is changed to sp^3.
To change a sp^3-hybrid to methane (break the bond of C-C) an energy greater than 3.20 eV is used in the program (3.60 eV do not work for incident-energies less then 20 eV). The chemical binding energy of C-H is 4.3 eV. It may be that this released energy reduces the energy necessary to break the single bonds.

The code has been enhanced to include the following chemical reactions:

1. If an impinging ion transfers more than a critical value of energy, ie. the energy-transfer $\geq$ 2.75 eV to a sp^2 carbon then the double bond is broken and a sp^3 carbon is created.

2. Since there is an upper limit of 1/3 on H atoms bonded to C for typical C:H film, this was used as a limit, *ie.*,
   - if the concentration of H (n_H) or sp^3 (n_{sp^3}) exceeds the concentration of sp^3_H ($n_{sp^3_H}$), sp^3 carbon is changed to sp^3_H.
   - if n_H or n_{sp^3} is less than $n_{sp^3_H}$, sp^3_H is transferred to sp^3.

3. The maximum concentration of carbon centers in sp^3 hybridization state (sp^3, sp^3_H) is controlled (as a feedback mechanism) by the values calculated by the Mech model [34] (sp^3,sp^3_H,sp^3_{CH}) and is dependent on the temperature. For example, at a given temperature, if the total concentration of carbon centers in sp^3 hybridization state calculated from SDTrimSP (n_{sp^3} + $n_{sp^3_H}$) is greater than the corresponding value predicted by the Mech model, then sp^3_H will be changed into sp^2 carbon.

Contribution from the kinetic part

The calculation of the kinetic part, $Y_0(E, T_0)$, of the erosion yield is based on the mechanism proposed by Hopf [33]. Let's say Y_{BB} represents the number of bond breaks of sp_H^3 caused by an impinging ion and $Y_{H_{stop}}$ gives the yield of the H atom implantation. These stopped or implanted H atoms can be bonded to free open carbon bonds.

For hydrocarbon formation both open carbon bonds and hydrogen atoms must be available. In order to include this effect the minimum of Y_{BB} and Y_{Hstop} is taken. For the out–diffusion probability of a hydrocarbon molecule from a depth x the kinetic part of the chemical erosion yield $Y_0(E, T_0)$ is given by

$$Y_0(E, T_0) = \int_0^d a_0 \cdot min[Y_{BB}, 2 \cdot Y_{H_{stop}}] \cdot e^{(-x/\lambda)}\ dx \qquad E > 3.2\ eV \tag{7.10}$$

Here, λ (= 0.4 nm) is the typical range of out-diffusion for hydrocarbon and d is the depth of the sample. If the whole sample is divided into k strips or depth intervals parallel to the surface as in SDTrimSP

$$Y_0(E, T_0) = \sum_0^k a_0 \cdot min[Y_{BB}(k), 2 \cdot Y_{H_{stop}}(k)] \cdot e^{(-x(k)/\lambda)}. \tag{7.11}$$

The minimum condition $min[Y_{BB}(k), 2 \cdot Y_{H_{stop}}(k)]$ expresses the fact that chemical erosion only happens when sufficient number of open bonds ($Y_{BB}(k)$) and sufficient number of hydrogen ($Y_{H_{stop}}(k)$) are available. Therefore, the minimum of both will limit the production of hydrocarbons. The value of the coefficient a_0 ranges from 0.03 to 1.0 and was obtained by the comparison of measurements and simulation results.

$$a_0 = min[1.0,\ 0.03 + 0.005 \cdot \frac{Y_{H_{stop}}(k)}{Y_{BB}(k)}] \tag{7.12}$$

Contribution from the thermal part

The temperature dependent part of the chemical erosion yield, $Y_T(E, T, \Phi_0)$, in the code is based on the Mech model [34].

$$Y_{TMech}(E, T, \Phi_0) = \varrho \cdot sp_{CH}^x \frac{Ex}{\Phi}\ \text{, where} \tag{7.13}$$

ϱ is the surface density of carbon atoms: $\varrho = 6.0 \cdot 10^{19}\ atoms/m^2$.

$$Y_T(E, T, \Phi_0) = \sum_1^n a_T \cdot Y_{T\_Mech}(E, T, \Phi_0) \cdot e^{(-x/\lambda)} \qquad E > 3.0\ eV \tag{7.14}$$

$$a_T = 1/(sp^3 + sp_{CH}^x)^{0.7} \tag{7.15}$$

Here n is the total number of collisions between ions and sp_H^3. A depth dependent out–diffusion probability after the hydrocarbon formation $e^{(-x/\lambda)}$ is added. The term $Y_{T\_Mech}(E, T, \Phi_0)$ is the yield of chemical erosion at a flux of ($\Phi_0 = 10^{18}\, m^{-2}s^{-1}$).

Flux dependence of chemical erosion
It is observed that the kinetic part of the erosion yield is flux independent. Therefore, in order to implement the correct flux dependence the thermal contribution (Y_{therm}) to the erosion has to be flux dependent. A flux interpolation is done with the help of Roth's formula [32].
The erosion yield given by the Mech model is correct only for a flux of $10^{18} m^{-2} s^{-1}$ (this is kept as the reference level flux Φ_0). It has been observed that with increasing flux, the temperature for which the yield is maximum ($T_{max}(E,\Phi)$) and the absolute value of the yield ($Y_{max}(E,\Phi)$) are increasing. It is assumed that $T_{max}(E,\Phi)$ as predicted by the Roth model is correct (although $Y_{max}(E,\Phi)$ is not). Then in order to calculate the flux dependent erosion yield for a given flux Φ and energy E, one proceeds as follows:

1. First the ΔT from Roth's formula (see Fig. 21) is computed

$$\Delta T = T_{max}(E,\Phi) - T_{max}(E,\Phi_0) \tag{7.16}$$

2. then the erosion yield from the Mech model at $T - \Delta T, \Phi_0$ is calculated

$$Y_{T\_Mech}(E, T - \Delta T, \Phi_0) \tag{7.17}$$

3. the corrected flux dependent erosion yield is given by

$$Y_{Tcor}(E,T,\Phi) = Y_{T\_Mech}(E, T - \Delta T, \Phi_0) \cdot \frac{max[Y_{Roth}(E,T,\Phi)]}{max[Y_{Roth}(E,T,\Phi_0)]} \tag{7.18}$$

If the sample is composed of n layers, then the total flux dependent thermal erosion yield is

$$Y_T(E,T,\Phi) = \sum_1^n a_\Phi \cdot Y_{Tcor}(E,T,\Phi) \cdot e^{(-x/\lambda)} \tag{7.19}$$

$$a_\Phi = 1/(sp^3 + sp^x_{CH})^{0.7} \quad . \tag{7.20}$$

Finally the total erosion yield of carbon is:

$$Y = Y_{physical}(E) + Y_0(E,T_0) + Y_T(E,T,\Phi) \quad . \tag{7.21}$$

In the following part of this section the general algorithm in the simulation and its technical implementation is presented, [35].
A pre–calculated number of energetic particles, depending on the fluence, are incident on the target. The target is divided into 1D layers and during each fluence step the incident particles initiate a collision cascade in the target. At the end of each fluence step the sputtering yield (chemical and physical), the scattering coefficient, and the implantation is calculated and then the sample is updated accordingly.

Depending upon the incident particle and target combination it is checked after each collision whether some thermal chemical reactions are possible or not. If yes, then the chemistry module is used to process the reactions. Then the depth dependent out–diffusion probability of the reactions products is also calculated and this is used to calculate the thermal part of the erosion yield.

During the cascade the number of broken bonds is calculated for each layer. During the collision with the target atoms the incident particles loose their energy and at the end of the collision cascade they get implanted into the target. The number of the implanted hydrogen atoms are calculated for each layer. At the end of the fluence step the total number of implanted hydrogen atoms and broken bonds are used for the calculation of the kinetic part of the erosion yield.

The maximum amount of bonded hydrogen has an upper limit of 1/3. At the end of each fluence step the relative concentration of the bonded hydrogen is calculated and if it exceeds the upper limit the surplus hydrogen is treated as freely moving. This hydrogen can then diffuse through the target and leave the system. The corresponding interchange among the different carbon hybridization states is also calculated simultaneously.

The surface binding energy (SBE) used for the pure carbon target (mainly sp^2 centers) is 4.00 eV.

When a sample of pure carbon is bombarded then sp^2 as well as sp^3 carbon centers are present and both of them have the same SBE of 4.00 eV. Finally all the counters for the bond breaking and hydrogen implantation in each layer are reset to zero.

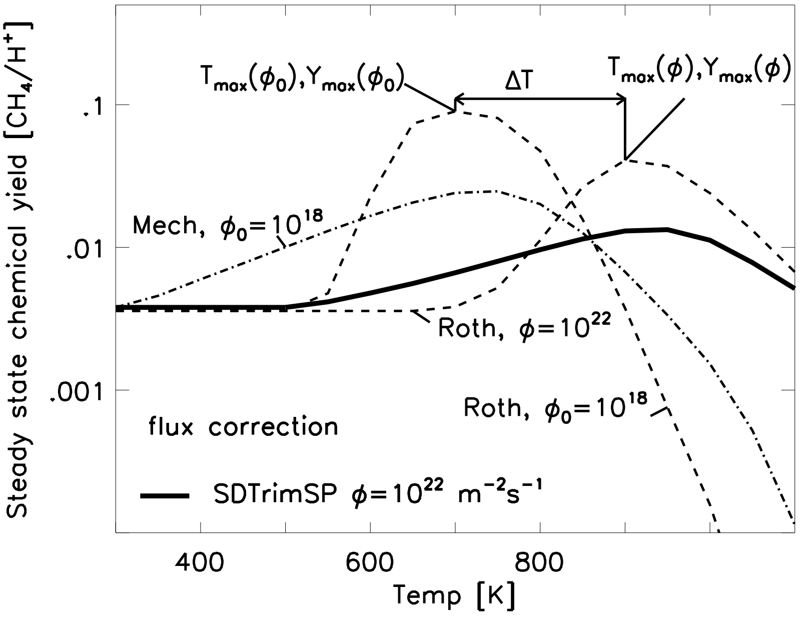

Figure 21: Correction of the thermal erosion yield of Mech [34] by a H flux $\Phi = 10^{18}\, m^{-2}s^{-1}$ to values for $\Phi = 10^{22}\, m^{-2}s^{-1}$ (SDTrimSP) using the analytical formula by Roth [32].

Comparison of calculated results of chemical erosion with experimental data

Fig. 22 shows a comparison of calculated results of H on C target with measurements at three different temperatures (T=300 K, 700 K and 800 K) as a function of the energy. The agreement of the calculated values with the experimental data is very good.
Fig. 23-25 shows a comparison of calculated methane yield of H on C target with measurements versus temperature at different energy and fluxes. The agreement is very good.
Fig. 26 provides a comparison of the calculated methane yield of 1000 eV H on a C target with measurements versus fluxes at different temperatures.

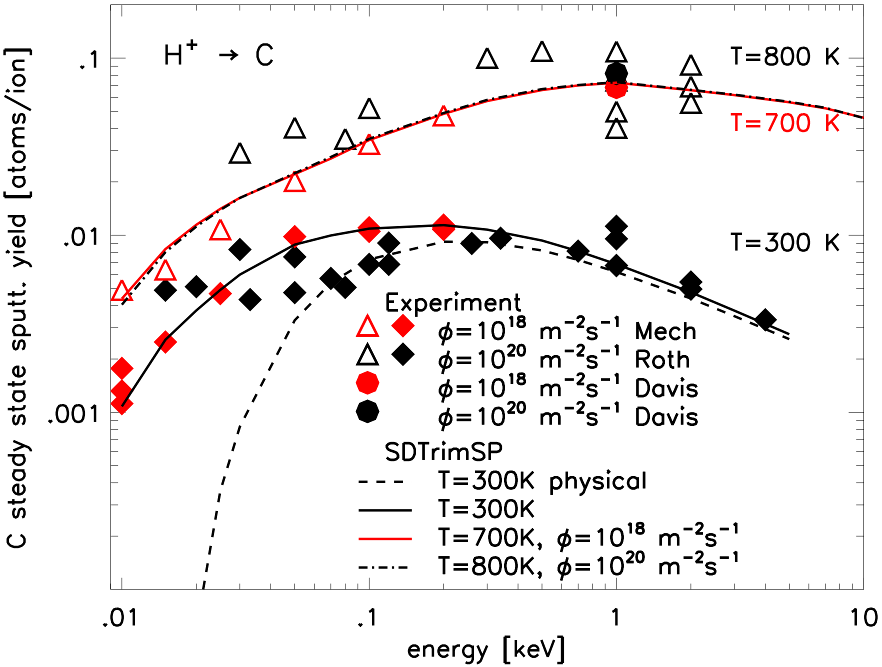

Figure 22: Calculated steady state sputtering yields of C versus the incident energy of H on a C target at normal incidence compared with experimental data [32], [34], [36].

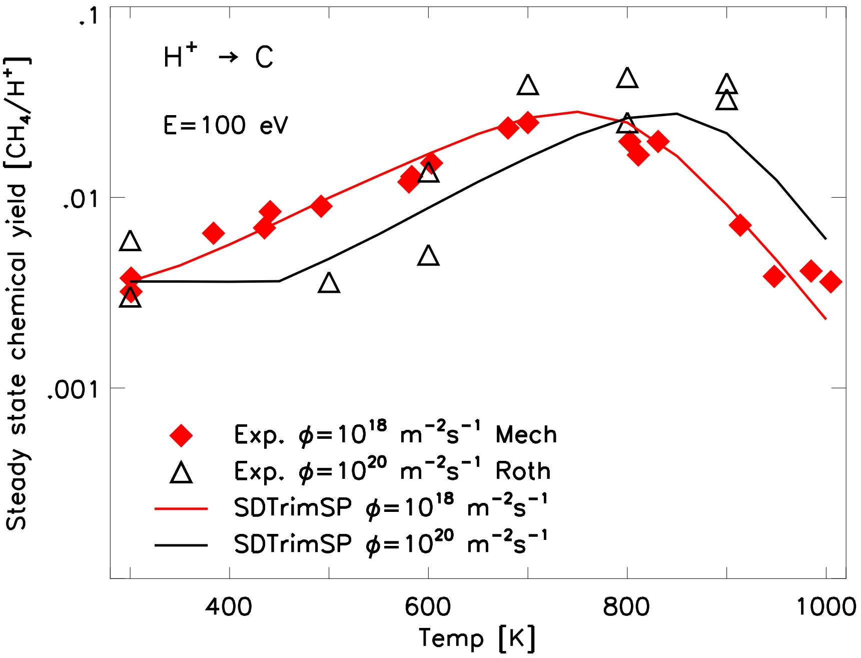

Figure 23: Calculated steady state sputtering yields of methane versus the temperature for 100 eV H on a C target at normal incidence for two different fluxes compared with experimental data [32], [34].

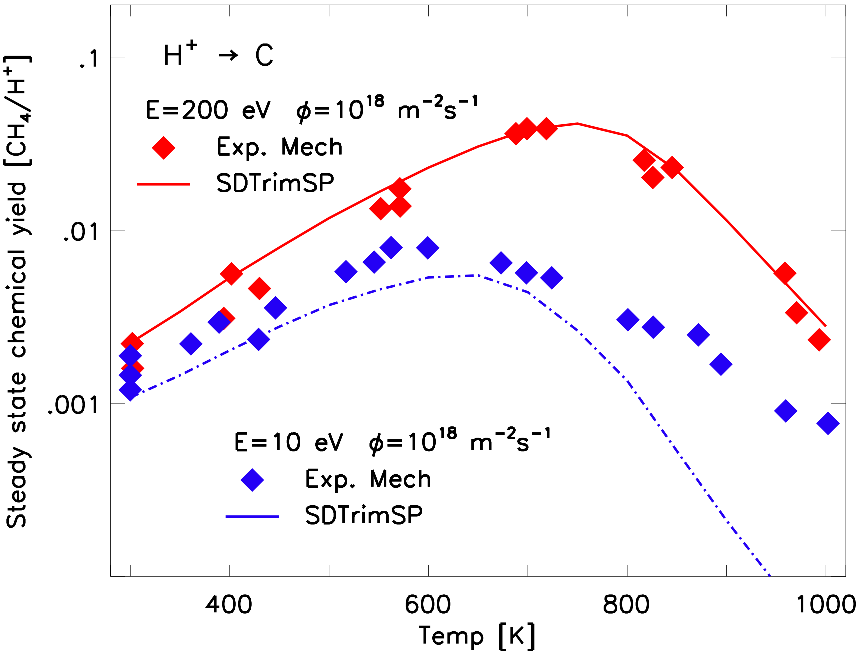

Figure 24: Calculated steady state sputtering yields of methane versus the temperature for 10 and 200 eV H on a C target at normal incidence compared with experimental data [34].

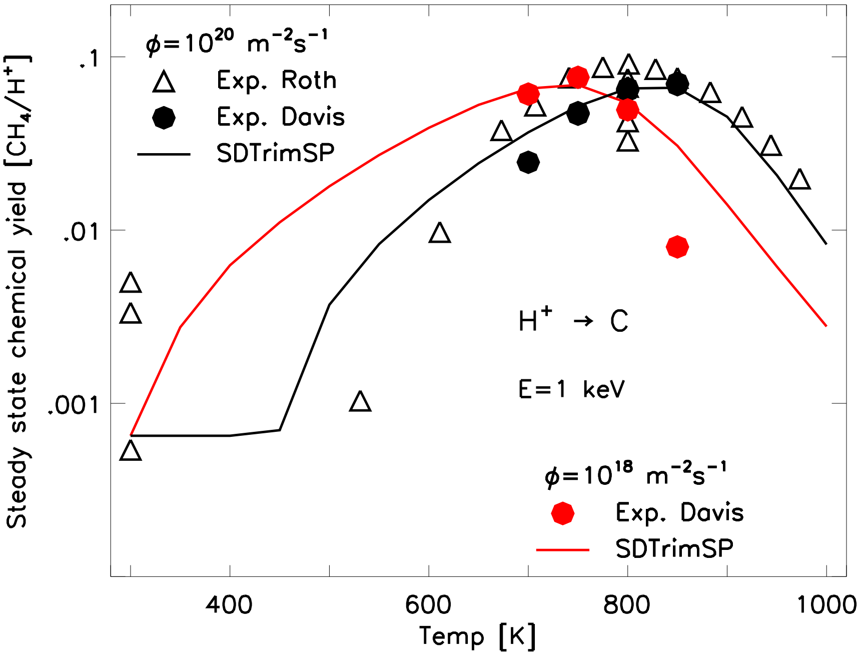

Figure 25: Calculated steady state sputtering yields of methane versus the temperature for 1000 eV H on a C target at normal incidence for two different fluxes compared with experimental data [32], [34], [36].

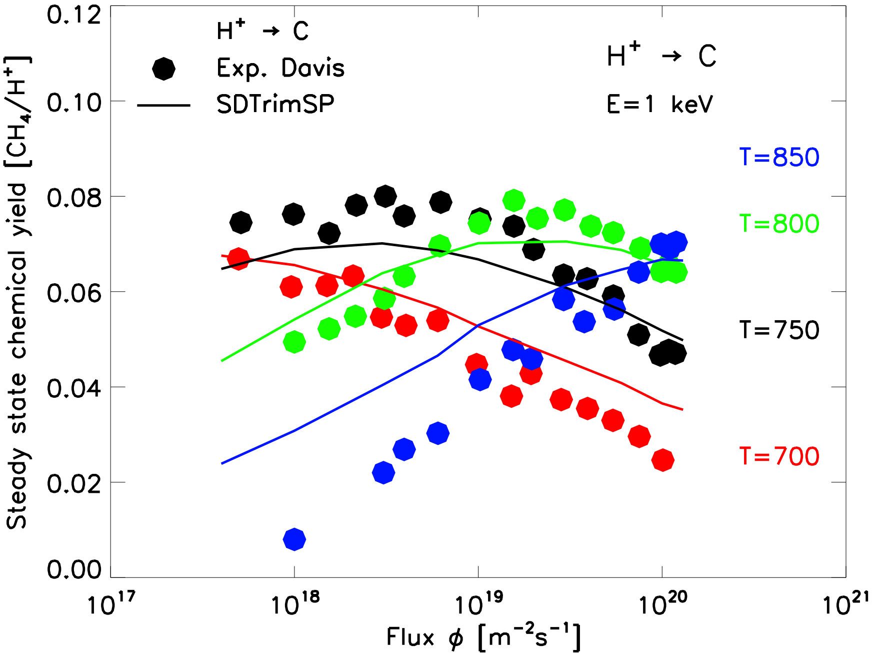

Figure 26: Calculated steady state sputtering yields of methane versus flux for 1000 eV H on a C target at normal incidence compared with experimental data [36].

Fig. 27 shows a comparison of calculated results of D on C target with measurements at two different temperatures as a function of the energy. Although the calculated results at room temperature are at the lower limit, the agreement is very good.

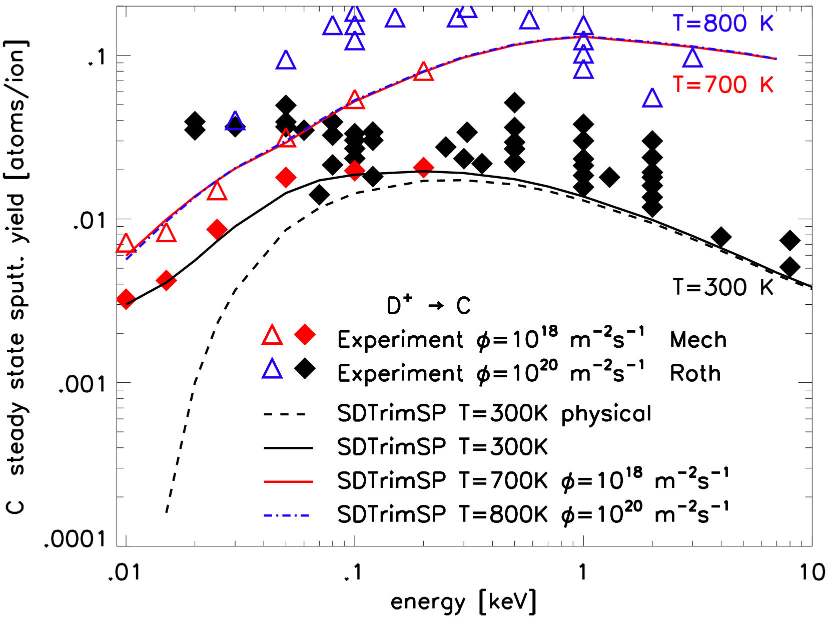

Figure 27: Calculated steady state sputtering yields of C versus the incident energy of D on a C target at normal incidence compared with experimental data [32].

7.4. New option for inelastic loss model (inel0=6)

To perform calculations with projectile energies that exceed ≈ 25keV / AMU the inelastic energy-loss by interaction with the target electrons requires a new stopping power model. To that end the stopping power model of Ziegler-Biersack [30] has been implemented in the new version of SDTRIM.SP. It allows to calculate the electronic stopping power of arbitrary ions in matter from eV to MeV energies. This goes beyond the already existing high energy stopping power models (inel = 4 or 5) which only applied to H or He projectiles. It requires the input parameters from "./tables/table6a" and "./tables/table6b" which are currently taken from SimNRA and have been determined by fitting a large database of stopping power data. To visualize the difference between the velocity proportional Lindhard-Scharff stopping power model (inel = 1) and the high energy stopping power model by Ziegler-Biersack (inel = 6), mean projected range calculations of 5 MeV C ions in W are compared in Fig. 28. This results in quite different range distributions showing the need of the new stopping power model for high energy calculations.

A typical example that can be modeled with the new inelastic loss model are the effects of high energy (MeV) ion implantation which is currently used as a proxy for high DPA neutron damage. In Fig. 29 the damage in W due to bombardment with 20MeV W ions is compared to an SRIM2013 [52] calculation. To improve the comparability

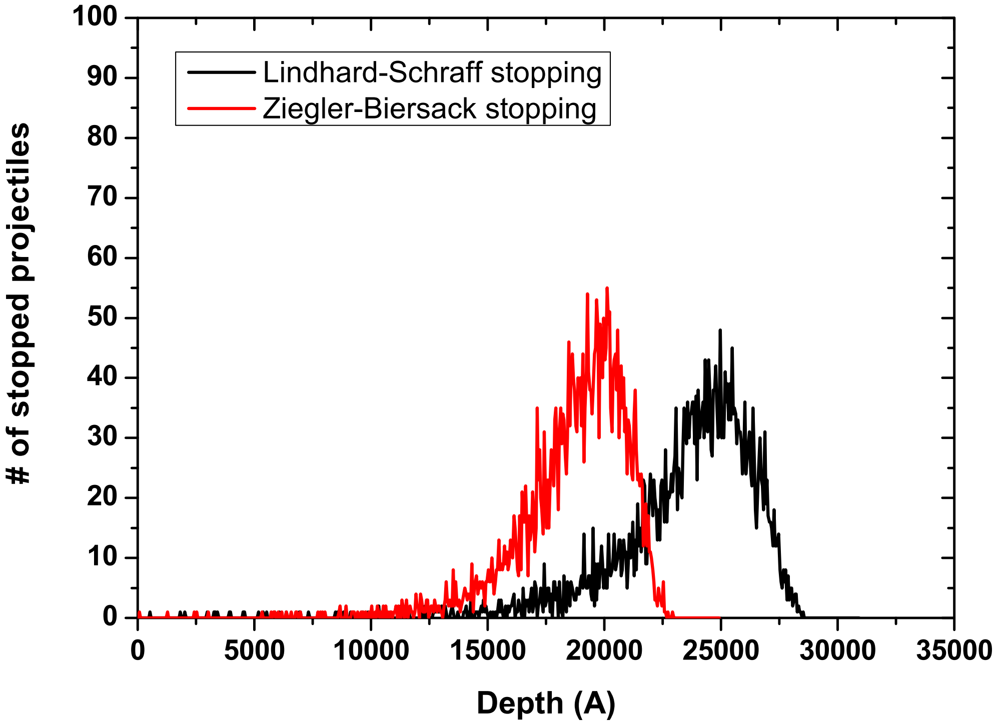

Figure 28: Comparison of implantation profiles of 5 MeV C in W calculated with Ziegler-Biersack and Lindhard-Scharff stopping

of damage calculations by SDTRIM.SP with those by the widely used SRIM code new counters and derived outputs were implemented in SDTRIM.SP. The rationale behind unifying the damage calculation output between the two codes is that the ASTM standard *Standard Practice for Investigating the Effects of Neutron Radiation Damage Using Charged-Particle Irradiation* [55] suggest to use SRIM in "Full Damage Cascade" mode to compute DPA values for comparing the damage produced in experiments performed with different projectiles.

It needs to be pointed out that the SRIM output in the full cascade mode violates energy conservation [54] and should therefore be grutinised.

The output of the new damage counters are added to the output files depth_proj.dat and depth_recoil.dat. In depth_proj.dat a new column named "NRT-DPA" was added. NRT-DPA stands for DPA values calculated according to the model by Norgett-Robinson-Torrens [53] which, after normalisation by the number of projectiles and layer width ($dx \times nh \times nr\_pproj$), is directly comparable to the SRIM output in file VACANCY.txt when operated in "Quick damage calculation" mode. In depth_recoil.dat three columns describe the damage produced during the cascade: "VACANCIES", "DISPLCNTR" and "REPLACEMENT". DISPLCNTR counts all displacement collisions (triggered by primary projectile and by recoils) where the transferred energy exceeds the displacement energy of the target atom. REPLACEMENT and VACANCIES are counters that subdivide displacement collisions in DISPLCNTR into two categories: In category A the incoming particle's energy after the collision is less that its own displacement energy and in category B it's remaining energy is greater than it's displacement energy. In category A, if the projectile and target atom are of the same species, the projectile replaces the target atom, this is recorded in the REPLACEMENT counter whereas in category B both the displaced recoil and the incoming projectile continue their trajectory and a vacancy is left behind, this is recorded in the VACANCIES counter. Again, after normalisation by the number of projectiles ($dx \times nh \times nr\_pproj$), the VACANCIES counter can be directly compared to the SRIM output in file VACANCY.txt and the REPLACEMENT counter to the output in NOVAC.txt when SRIM is operated in "Full Damage Cascade" mode. In Fig. 29 the calculation results of SRIM and SDTRIM.SP of a self-damage experiment are compared. In this self-damage experiment a W-target is bombarded by 20MeV W ions to mimic the effect of fusion neutron irradiation.

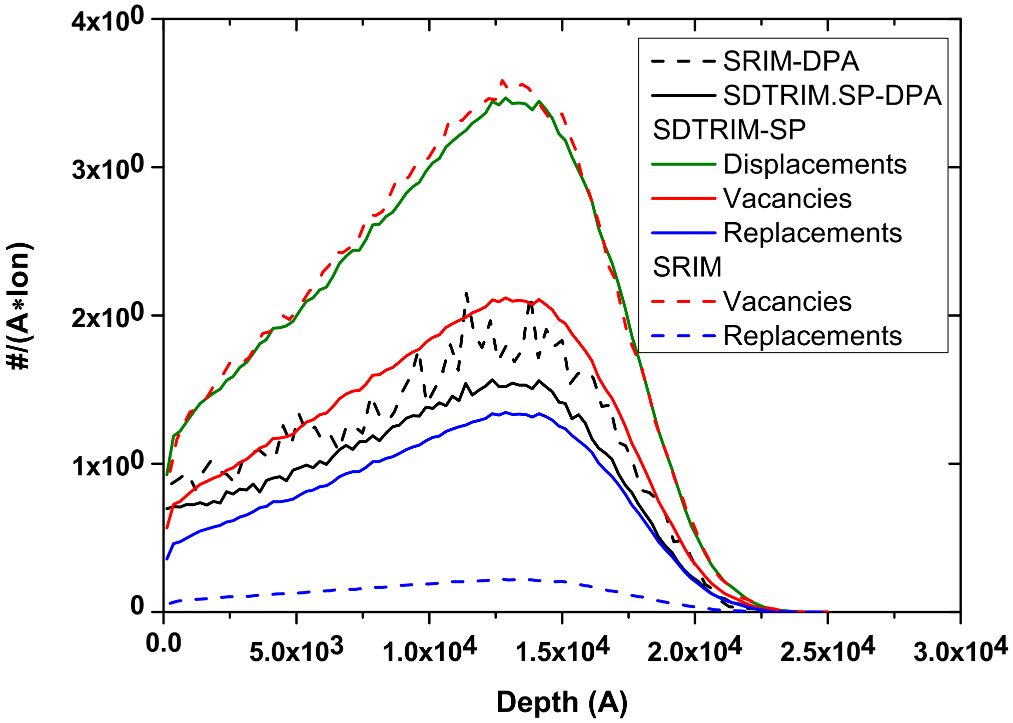

Figure 29: Comparison of radiation damage due to MeV self implantation W between SDTRIM.SP and SRIM2013

Comparing the output of SRIM with SDTRIM.SP in Fig. 29 shows that the DPA values computed by the NRT model in SDTRIM.SP match well the values from SRIM. Comparing the vacancies and replacements shows that SRIM yields less replacement collisions than SDTRIM.SP but yields more vacancies. The SRIM computed vacancies match the total number of displacements computed by SDTRIM.SP, which given the fact that replacements + vacancies ≈ displacements is to be expected. In conclusion both codes compute the same number and depth distribution of displacement events but differ in the sub-division into vacancies and replacements. Due to the closed source nature of SRIM the origin of this difference cannot be determined.

7.5. Thermal diffusion (lterm_dif, a_0, e_act)

There the solid-steate diffusion within the target is modelled (not outgasing). It typically matters only for high temperatures and low impinging fluxes. It is assumed that the diffusion flux is proportional to the concentration gradient.
The thermal diffusion depends on the temperature T, the time t, the atomic fraction qu and coefficients a_0 and e_act. The coefficients a_0 and e_act are matrices, which depend on the species in the target. The diffusion equation

$$\frac{\partial n(x,i)}{\partial t} = \frac{\partial(\eta(i) \cdot \frac{\partial n(x,i)}{\partial x})}{\partial x}$$

with the defusion coefficient $\eta(i)$ given by

$$\eta(i) = a\_0(i,i) \cdot e^{-e\_act(i,i)/kT}$$

The boundary condition at the surface is:

$$\frac{\partial n}{\partial x}|_{x-0} = 0$$

The time t needed for a fluence step Δflc depends on the flux of projectiles and is given by

$$t = \frac{\Delta flc}{flux}$$

The flag for diffusion is lterm_dif and the names of the input-files are ”a0_tdiff.inp” and ”eact_tdiff.inp”, see appendix.
default: **lterm_dif=.false., a_0(:,:)=0.0, e_act_0(:,:)=0.0**

The method is controlled with flag i_diff_algo, see appendix.
default: **i_diff_algo=2**

i_diff_algo	=0 :	simple diffusion
	=1 :	explicit algorithm (lambda_cn=0)
	=2 :	implicit algorithm (lambda_cn=1)
	=3 :	Crank-Nicolson (lambda_cn=0.5)
	=4 :	Crank-Nicolson with lambda_cn=0...1

The full implicit algorithm for diffusion equation is slightly faster and stable than the Crank-Nicolsen thus the recommended setting.

An example of the full coefficient-matrix for a C-W target found in [37]. All diffusion-coefficients (C in W, C in C, W in C and W in W) are same.

$$a\_0 = \begin{pmatrix} 8.91 & 8.91 \\ 8.91 & 8.91 \end{pmatrix} \cdot 10^{-6}\ [m^2/s]$$

$$e\_act = \begin{pmatrix} 2.32 & 2.32 \\ 2.32 & 2.32 \end{pmatrix}\ [eV]$$

In version 6.00 is for one species (for example C) no distinction between C in W or C in C. This means only the values from the diagonal are used.

$$a\_0 = \begin{pmatrix} 8.91 & 0.00 \\ 0.00 & 8.91 \end{pmatrix} \cdot 10^{-6}\ [m^2/s]$$

$$e\_act = \begin{pmatrix} 2.32 & 0.00 \\ 0.00 & 2.32 \end{pmatrix}\ [eV]$$

Fig. 30 shows the time dependend profile of a C-layer in a W-target.

Another example of the coefficient-matrix is a bombardment of a Fe-W target with D, see Fig. 31. Only Fe and W can diffuse.

$$a\_0 = \begin{pmatrix} 0.00 & 0.00 & 0.00 \\ 0.00 & 1.5e-2 & 0.00 \\ 0.00 & 0.00 & 1.5e-6 \end{pmatrix}\ [m^2/s]$$

$$e\_act = \begin{pmatrix} 0.00 & 0.00 & 0.00 \\ 0.00 & 2.97 & 0.00 \\ 0.00 & 0.00 & 2.97 \end{pmatrix}\ [eV]$$

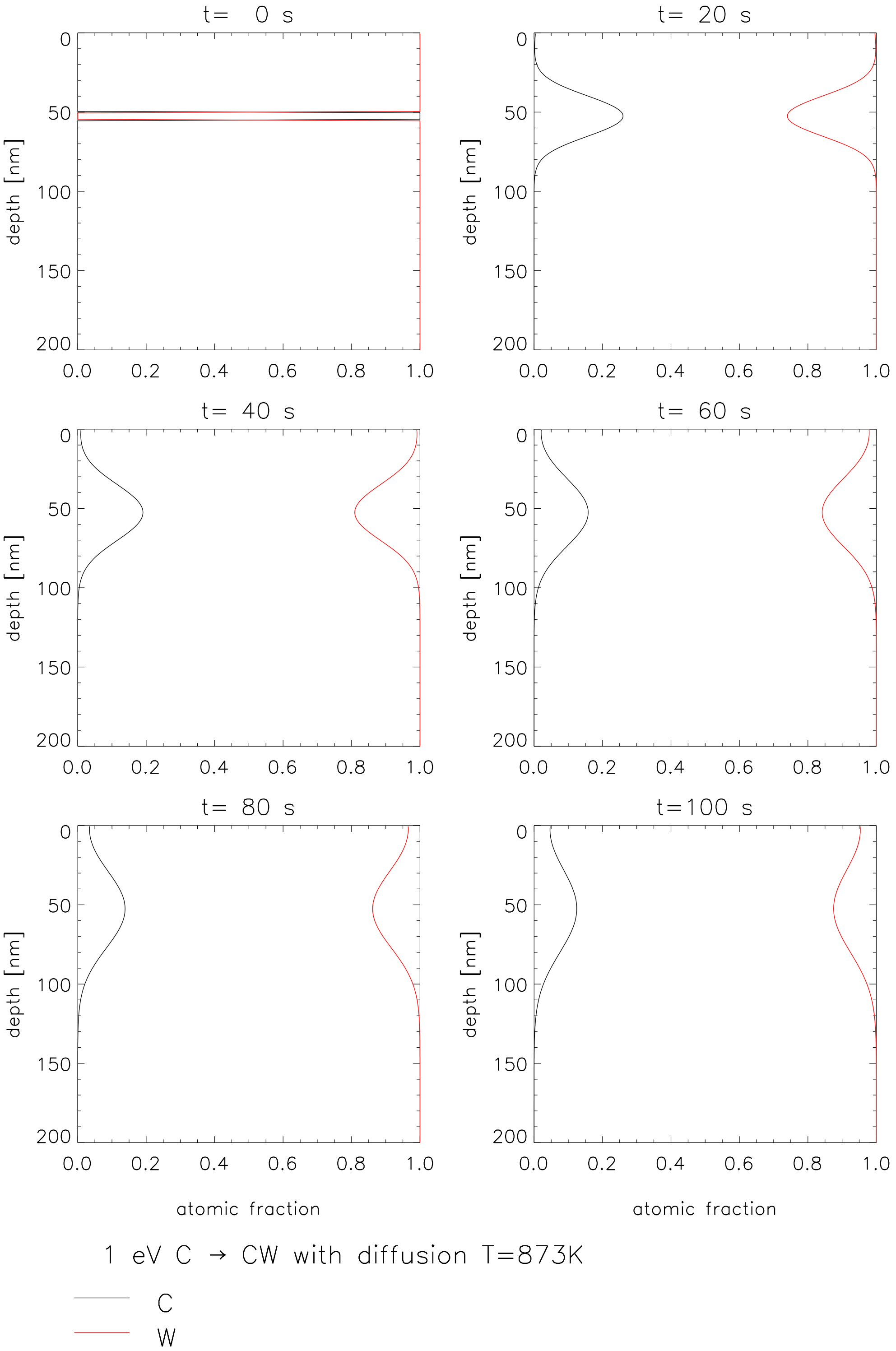

Figure 30: Calculated thermal diffusion of a thin C-layer in a W target.

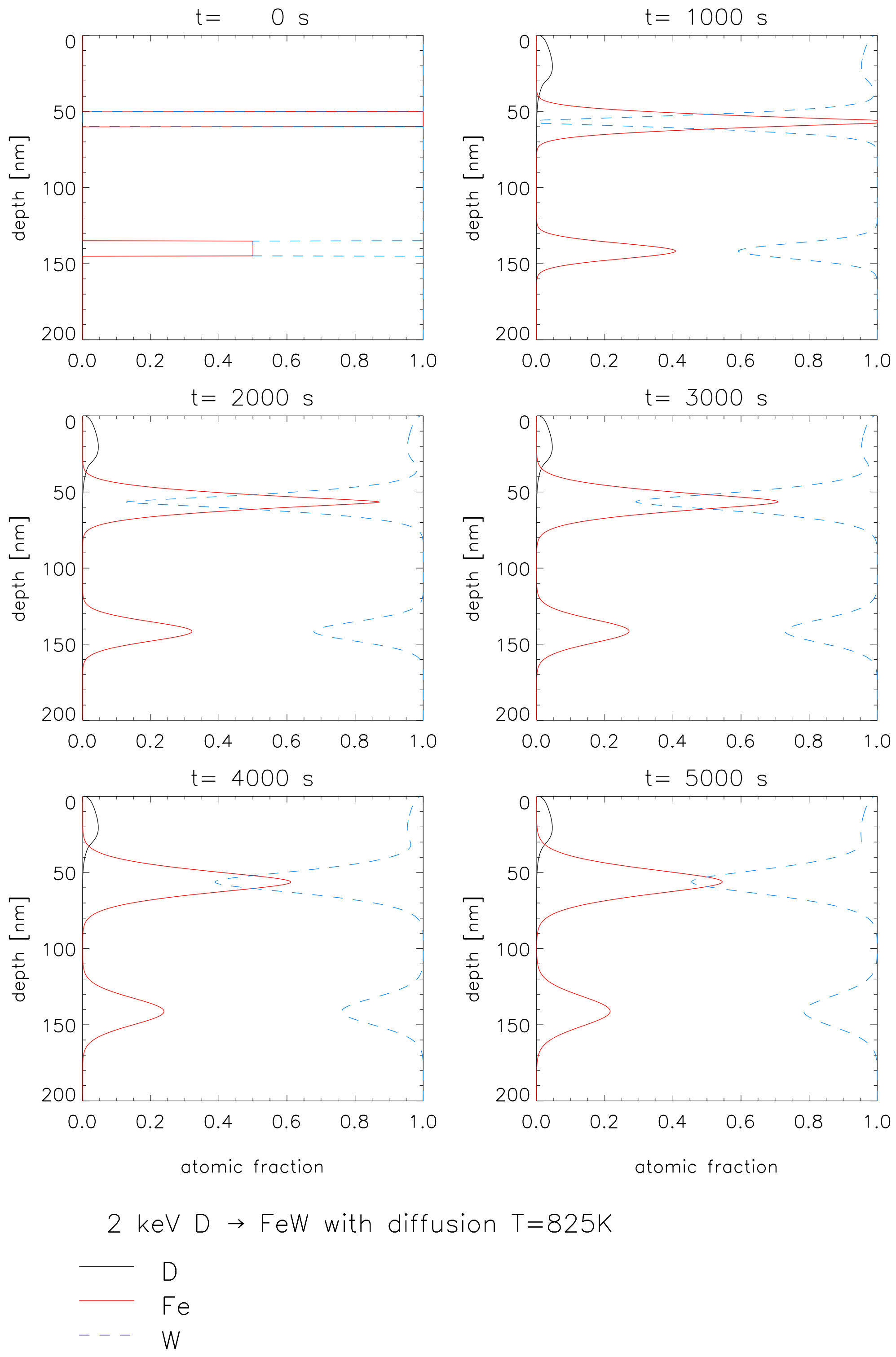

Figure 31: Calculated thermal diffusion of thin Fe-layers in a W target.

7.6. New surface-binding-model 7

Following the paper of Kudriavtsev et al. [38] a new option **isbv=7** of the surface binding energy was implemented. Surface binding energy (E_s) is the energy needed to extract one atom from the top surface during ion sputtering process to the vacuum. The model described by Kudriavtsev is a theoretical approach, that only needs minor input data to calculate the E_s.

$$E_s = t_c \cdot E_{covalent} + t_i \cdot E_{ionic} \tag{7.22}$$

with the weighting coefficients: $t_c = (1 - t_i)$

E_s can be calculated, if the electronegativity S, the covalent radii r_{cov} and the covalent binding energy E_c are known for the single elements. Let T be the index of the target atom and A the index of the sputtered element. In the following the calculation of the covalent and the ionic binding energy is shown in detail.
The first term in 7.22 represents covalent binding energy.

$$E_{covalent} = E_c^{A-T} \cdot \frac{(n/2) r_{cov}^{A-T}}{d}$$

$$\begin{aligned} \text{with: } E_c^{A-T} &= \sqrt{E_c^A \cdot E_c^T} \\ n/2 &= 1.4 \\ r_{cov}^{A-T} &= r_{cov}^A + r_{cov}^T \\ d &= r_{cov}^A + r_{cov}^T - c_x \cdot |S_A - S_T| \\ c_x &= 0.007\ nm \end{aligned}$$

The second term in equation 7.22 is the ionic binding energy, known as coulomb potential for two separate charges reduced by the partial charge Δq. ϵ_0 is the electric field constant and e the electron charge.

$$E_{ionic} = \frac{e^2}{d \cdot 4\pi \cdot \epsilon_0}$$

$$\Delta q \ = 0.16 \cdot |S_A - S_T| + 0.035 \, |S_A - S_T|^2$$

Finally the equation for the surface binding energy is:

$$E_s = (1 - \Delta q) \cdot E_c^{A-T} \cdot \frac{(n/2) r_{cov}^{A-T}}{d} + \Delta q \cdot \frac{e^2}{d \cdot 4\pi \cdot \epsilon_0} \tag{7.23}$$

Treatment of mixed target compounds
The model can be used for pure targets, as shown before or for targets with more species. Only minor changes have to be applied.
The covalent binding energy of nuclear molecules, has to be computed for the target itself and afterwards for the system of target and projectile. Let be T_1 the index of first target atom, T_1 the index of second target atom and A the index of the projectile. The electronegative of the target is calculated via geometric mean of its consisting species:

$$\begin{aligned} E_c^T &= \sqrt{E_c^{T_1} \cdot E_c^{T_2}} \\ r_{cov}^T &= r_{cov}^{T_1} + r_{cov}^{T_2} \\ S_T &= \sqrt{S_{T_1} \cdot S_{T_2}} \end{aligned} \tag{7.24}$$

The covalent radius of the target is calculated as sum of the corresponding covalent radii. Different stoichiometric factors are not considered in the model.

One example is the bombardment of SiO_2 with Ar (use the equation 7.23 and 7.24):

$Es(Ar)$	:	index A = "Ar",	index T_1 = "Si" and	index T_2 = "O"
$Es(Si)$	:	index A = "Si",	index T_1 = "Si" and	index T_2 = "O"
$Es(O)$	:	index A = "O",	index T_1 = "Si" and	index T_2 = "O"

An other example is the bombardment of Si with Ar:

$$
\begin{array}{lll}
Es(Ar) & : \text{ index A} = \text{"Ar"}, & \text{index } T_1 = \text{"Si"} \\
Es(Si) & : \text{ index A} = \text{"Si"}, & \text{index } T_1 = \text{"Si"}
\end{array}
$$

Comparison of surface binding energies for silicon

As an example, the surface binding energy of pure silicon for different elements was calculated.

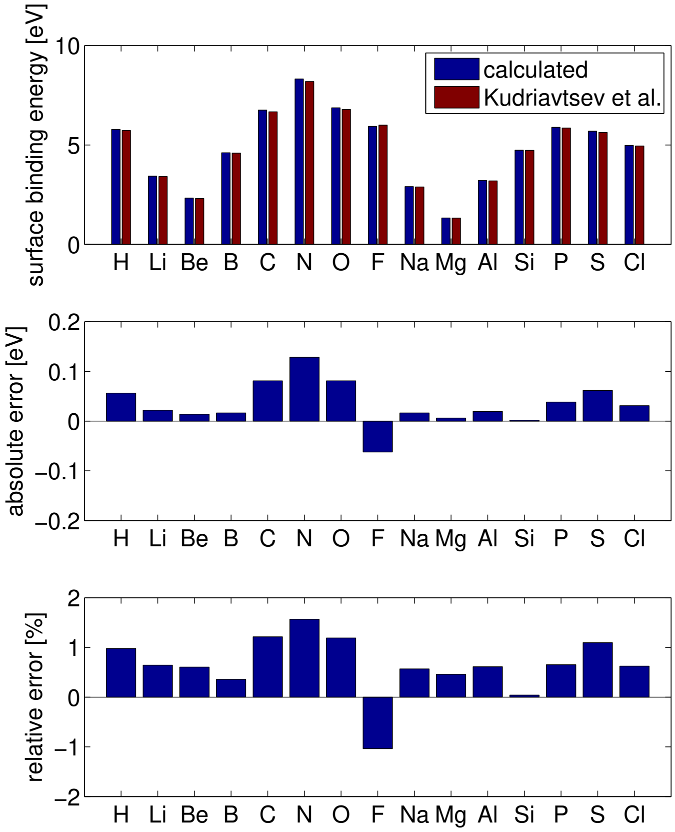

Figure 32: Surface binding energies for different elements on silicon (blue) calculated values without further fitting (red) values from Kudriavtsev hhh

With a fitting parameter c_x of 0.007 the difference between the paper values and the calculated ones is below (±)2%.

8. Sensitivity studies for selected examples

8.1. Number of incident particles

In a Monte Carlo code the number of test particles defines the accuracy of the results. Fig. 33 shows the value of the sputtering yield for D on Be at normal incidence for three different energies as a function of the number of incident D particles. Fig. 34 shows the relative error of the yields for this example. In this particular case the number of incident projectiles should be more than 10^6 particles to obtain sufficiently good statistic estimates.

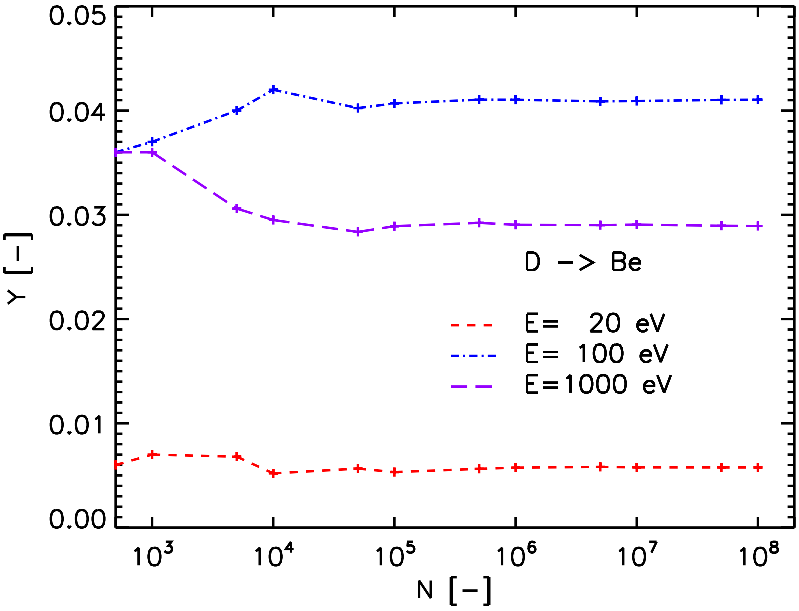

Figure 33: Calculated absolute yield of Be depending on number of incident particles of D on a Be target at normal incidence for three different energies.

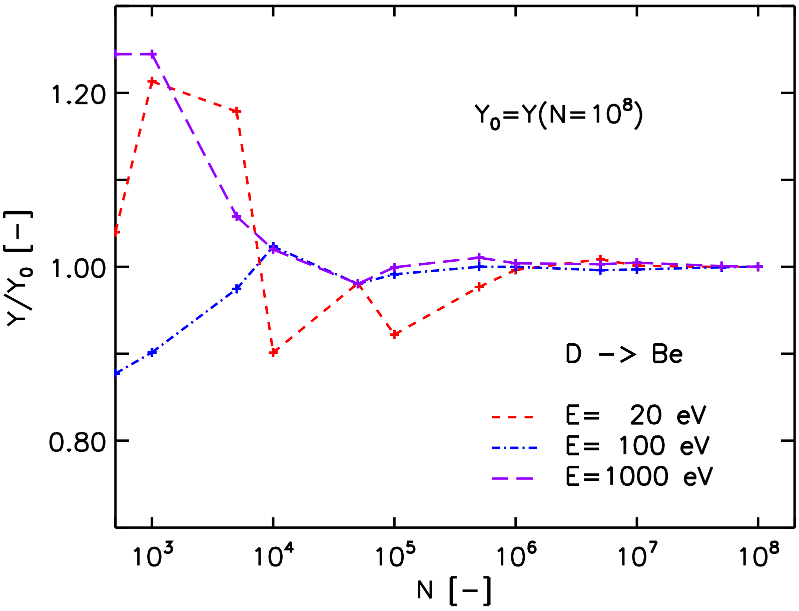

Figure 34: Relative deviation Y to Y_0 a function of the number of incident D particles on a Be target at normal incidence for three different energies

8.2. Integration methods (iintegral)

The scattering angle ϑ in the center-of-mass system is determined by [1]

$$\vartheta = \pi - 2p \int_R^\infty r^{-2} g(r)^{-1} dr \tag{8.25}$$

$$g(r) = \sqrt{1 - \frac{p^2}{r^2} - \frac{V(r)}{E_r}}\ , \quad g(R) = 0 \tag{8.26}$$

where p is the impact parameter, R is the distance of closest approach of the two collision partners, $V(r)$ is the interaction potential and R_r is the energy in the center-of-mass system.

The other important integral is the time-integral τ

$$\tau = \sqrt{r^2 - p^2} - \int_R^\infty g(r)^{-1} - f(r)^{-1} dr \tag{8.27}$$

$$f(r) = \sqrt{1 - \frac{p^2}{r^2}} \tag{8.28}$$

The numerical solution of the integrals 8.25 and 8.27 is possible with the Gauss-Mehler quadratur [14], the Gauss-Legendre quadratur [1] or the 'magic' algorithm from Biersack [1]. The results of the different integration methods are shown in Fig. 35 and Fig. 36.
All calculation were made with the '$Kr - C$' Potential and surface-binding-model three. The difference between the integration methods of Gauss-Mehler and Gauss-Legendre is small.

The method 'MAGIC' is only an approximate method and it agrees well only for high energy ranges. It should be avoided and is provided for backward compatibility used.

Default is the use of the integral-method Gauss-Legendre-Quadrature: **iintegral=2**
This method use a number of pivot-coefficients. Default is: **ipivot=8**

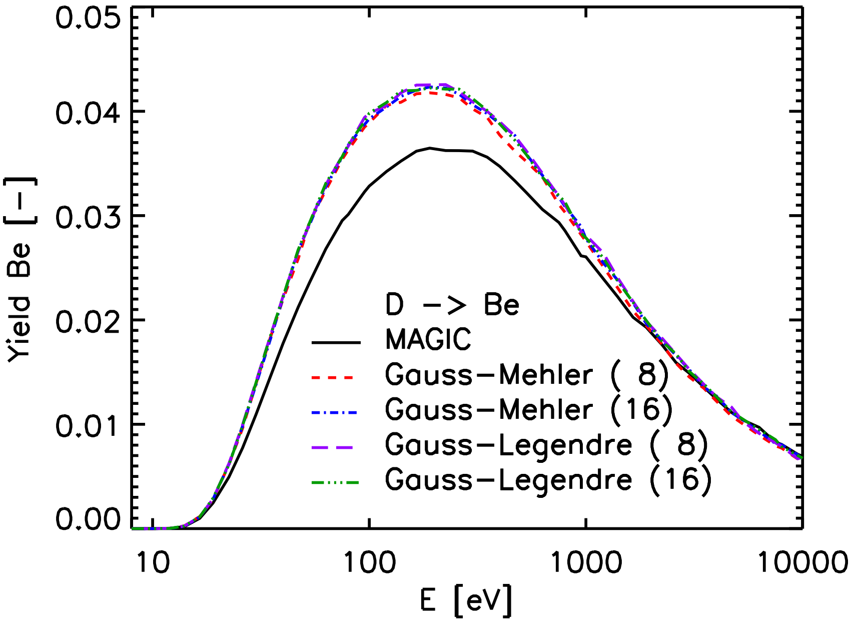

Figure 35: Calculated sputtering yield Y D on a Be target for three different integration methods. Gauss-Mehler and Gauss-Legendre were calculated with 8 and 16 numbers of integration-steps (ipivot).

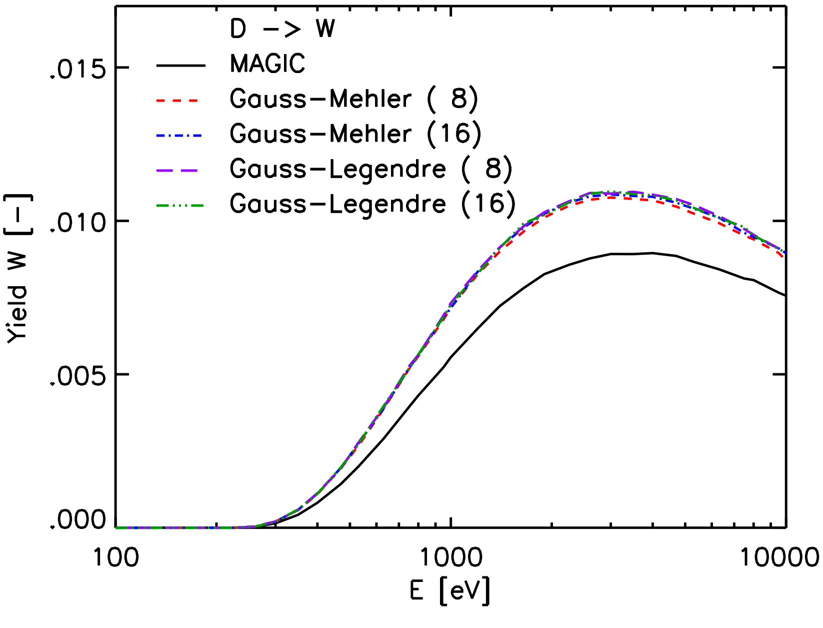

Figure 36: Calculated sputtering yield Y for D on a W target for three different integration methods. Gauss-Mehler and Gauss-Legendre were calculated with both 8 and 16 numbers of integration-step (ipivot).

8.3. Screened coulomb potentials

The reduced energy ε is:

$$\varepsilon = a \cdot \frac{M_2}{Z_1 \cdot Z_2 \cdot e^2 \cdot (M_1 + M_2)} \cdot E \tag{8.29}$$

with M_1 relative atomic mass of projectile, M_2 relative atomic mass of target atom. Often used for calculation of ε is the screening length a following Lindhard-Scharff, see [1].

$$a = a_{LS} = ab_{bohr} \cdot \frac{p_{const}}{Z_1^{2/3} + Z_2^{2/3})^{-1/2}} \qquad \text{(Lindhard-Scharff)} \tag{8.30}$$

The screened coulomb potentials $V(r)$, see [1], can be written in the form

$$V(r) = \frac{Z_1 \cdot Z_2 \cdot e^2}{r} \cdot \Phi(\frac{r}{a}) \tag{8.31}$$

with Z_1 atomic number of projectile, Z_2 atomic number of target atom, a the screening length and r the distance between the two atoms.

The equation for different interaction potential are given below:
for the **Krypton-carbon** potential:

$$\Phi(\frac{r}{a}) = 0.191 \cdot e^{-0.278r/a} + 0.474 \cdot e^{-0.637r/a} + 0.335 \cdot e^{-1.919r/a} \tag{8.32}$$

$$a = a_F = ab_{bohr} \cdot \frac{p_{const}}{\sqrt{Z_1} + \sqrt{Z_2})^{2/3}} \qquad \text{(Firsow)} \tag{8.33}$$

for the **Moliere potential**:

$$\Phi(\frac{r}{a}) = 0.35 \cdot e^{-0.3r/a} + 0.55 \cdot e^{-1.2r/a} + 0.10 \cdot e^{-6.0r/a} \tag{8.34}$$

$$a = a_F = ab_{bohr} \cdot \frac{p_{const}}{\sqrt{Z_1} + \sqrt{Z_2})^{2/3}} \qquad \text{(Firsow)} \tag{8.35}$$

for the **ZBL** potential:

$$\begin{aligned} \Phi(\frac{r}{a}) &= .028171 \cdot e^{-0.20162r/a} + .28022 \cdot e^{-0.4029r/a} + .50986 \cdot e^{-0.94229r/a} \\ &+ .18175 \cdot e^{-3.1998r/a} \end{aligned} \tag{8.36}$$

$$a = a_{ZBL} = ab_{bohr} \cdot \frac{p_{const}}{(Z_1^{0.23} + Z_2^{0.23})} \qquad \text{(Ziegler,Biersack,Littmark: ZBL)} \tag{8.37}$$

for the **Nakagava-Yamamura** potential:

$$\Phi(\frac{r}{a}) = e^{(-A\cdot(r/a)+B\cdot(r/a)^{1.5}-C\cdot(r/a)^2)} \tag{8.38}$$

with:

$$\begin{aligned} A &= 1.51 \\ B &= 0.763 * (Z_1^{0.169} + Z_2^{0.169})/(Z_1^{0.307} + Z_1^{0.307}) \\ C &= 0.191 * ((Z_1^{0.0481} + Z_2^{0.0481})/(Z_2^{0.307} + Z_2^{0.307}))^{4./3.} \end{aligned} \tag{8.39}$$

$$a = a_{NY} = ab_{bohr} \cdot \frac{p_{const}}{(Z_1^{0.307} + Z_2^{0.307})^{2/3}} \qquad \text{(Nakagava-Yamamura)} \tag{8.40}$$

Name	value	unit	
ab_{bohr}	0.52917725	[Å]	Bohr radius
p_{const}	0.885341377	[-]	$(9 \cdot \pi^2/128)^{1/3}$
e^2	14.399651	[eV Å]	

Table 4: constants

Fig. 37 and Fig. 40 show the results of calculation of four different potentials. It is hard to say which potential is the best. The Kr-C potential seems good for all examples. All calculations were done with Gauss-Legendre integration, surface-binding-model one ($isbv = 1$) and inelastic loss model from Lindhard-Scharff ($inel = 1$). The difference between the potentials is relativly small.

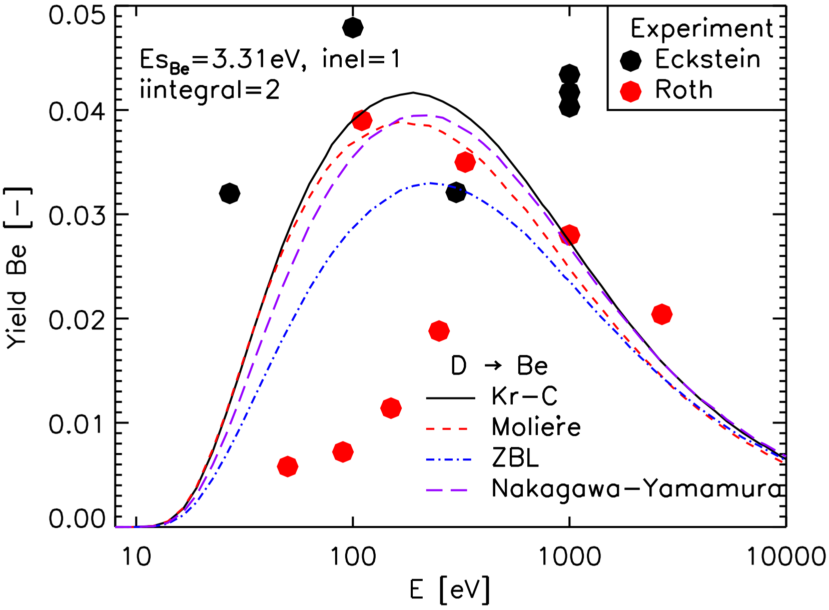

Figure 37: Calculated and measured sputtering yield Y of Be [28], [29] , D on a Be target for four different potentials.

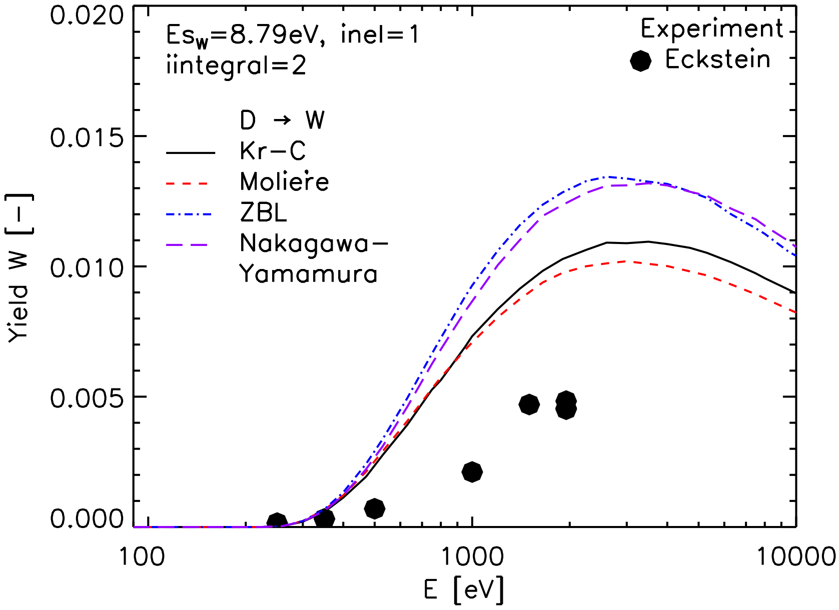

Figure 38: Calculated and measured sputtering yield Y of W [28], D on a W target for four different potentials.

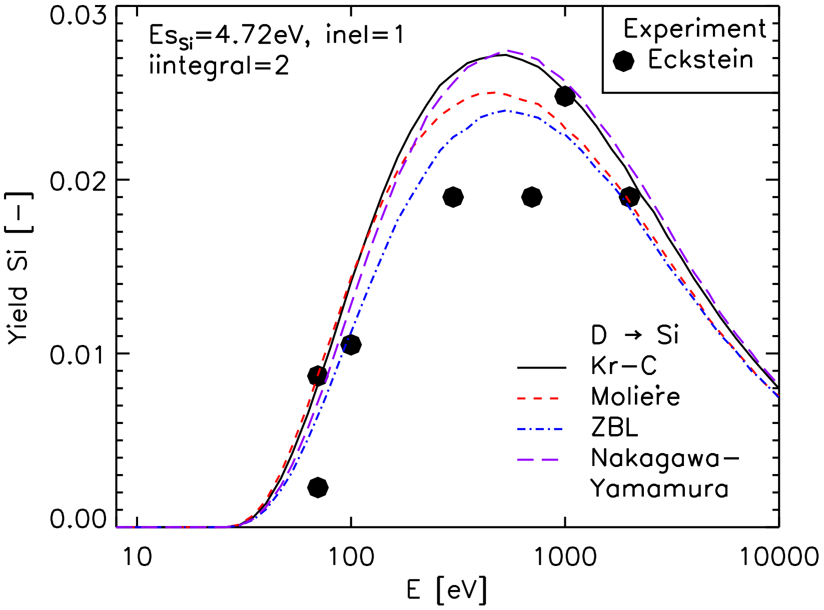

Figure 39: Calculated and measured sputtering yield Y of Si [28], D on a Si target for four different potentials.

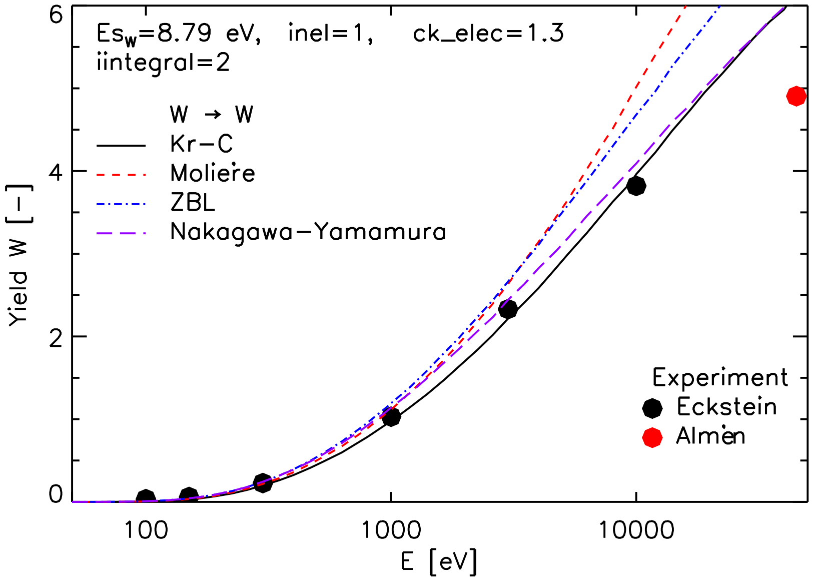

Figure 40: Calculated and measured sputtering yield Y of W [28], [50], W on a W target for four different potentials.

8.4. Surface binding energy

The user can choose from 7 methods ($isbv$) to calculate the local surface binding energy (sbe) of each species in a target. Es is the atomic surface binding energy for each species and q the atomic fraction at surface.

8.4.1. D on W

For a static calculation $q_D = q_1 = 0$ and $q_W = q_2 = 1$. In model one Es is independent of the composition of the target-surface. In model two and three Es is instead dependent on the surface composition.

$$\begin{aligned} Es_1 = Es_D &= 1.10\ eV \\ Es_2 = Es_W &= 8.79\ eV \\ Es_{1,2} = Es_{DW} &= 0.5 \cdot (1.10 + 8.79) = 4.945\ eV \end{aligned}$$

$$\begin{aligned} \textbf{model 1: } (isbv = 1)\ sbe_D &= Es_1 = 1.10\ eV \\ sbe_W &= Es_2 = 8.79\ eV \end{aligned} \tag{8.41}$$

$$\begin{aligned} \textbf{model 2: } (isbv = 2)\ sbe_D &= q_1 \cdot Es_1 + q_2 \cdot Es_2 = 8.79\ eV \\ sbe_W &= q_1 \cdot Es_1 + q_2 \cdot Es_2 = 8.79\ eV \end{aligned} \tag{8.42}$$

$$\begin{aligned} \textbf{model 3: } (isbv = 3)\ sbe_D &= q_1 \cdot Es_1 + q_2 \cdot Es_{1,2} = 4.945\ eV \\ sbe_W &= q_1 \cdot Es_{1,2} + q_2 \cdot Es_2 = 8.79\ eV \end{aligned} \tag{8.43}$$

Only the surface-binding energy of D is changed, because $q_D = 0$. Therefore only the scattering coefficient depends on the methods, see Fig. 41. All calculations used the '$Kr - C$' potential and the integration-method of 'Gauss-Legendre'.

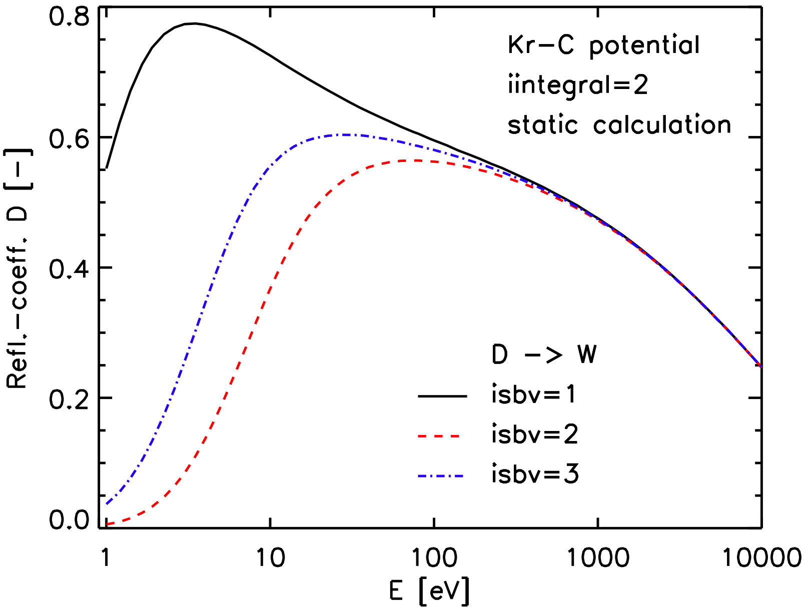

Figure 41: Calculated reflection coefficient depending on surface binding model of D on a W target at normal incidence in static mode.

8.4.2. Ar on Ta_2O_5

Fig. 43 shows the comparison of results Ar on Ta_2O_5 for a static (idrel=1) and dynamical (idrel=0) calculation. If the target composition at the surface is not constant (preferential sputter) then it is necessary to use the dynamical mode.
The Values for the case Ar on Ta_2O_5 are:

$$\begin{aligned} Es(O) &= 2.582\ eV = 0.5 \cdot \Delta H_{diss}(O_2) \\ \Delta H_{diss}(O_2) &= 5.164\ eV \\ Es(Ta) &= \Delta H_S(Ta) = 8.100\ eV \\ \Delta H_f(Ta_2O_5) &= 21.2053\ eV \end{aligned}$$

Usually the heat of sublimation ΔH_S or heat of dissociation ΔH_{diss} for a gas are used for the surface-binding-energy Es. Comparisons with measurements show that it is better to use for oxygen $Es = 1.0\ eV$
The target is not constant. The calculated static values of the sputtering coefficients are wrong. Therefore it necessary to use the dynamical mode for calculations of yields, see Fig. 42.

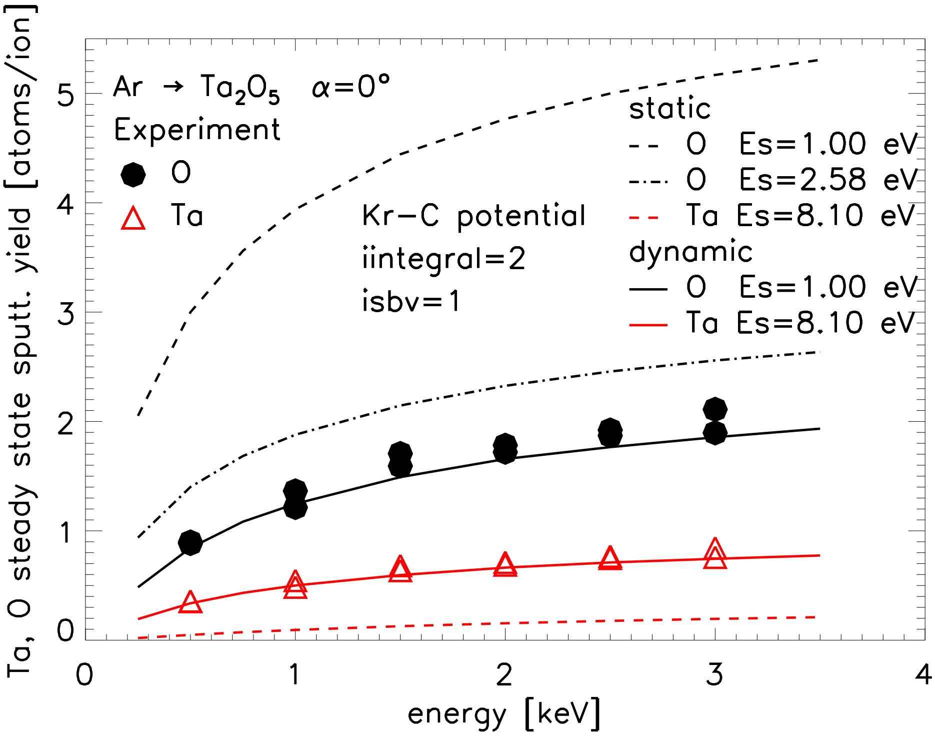

Figure 42: Comparison of sputtered yields calculated for static and dynamical mode (idrel=1,0) with experimental results [40], Ar on Ta_2O_5.

Fig. 43 shows the comparison of different surface-binding-models *isbv* with measurements by bombardment of Ta_2O_5 with Ar, [40]. The target in this example is a solid-gas compound. The special model is the model 5 (isbv=5) for this case and use the heat of formation ΔH_f for the calculation of Es, [39].
The surface-binding-energy for the different models (isbv) are:

$$\begin{aligned}\text{isbv} \quad &= 1: Es(Ta) = 8.1\ eV \quad Es(O) = 1.0\ eV \quad \text{(use eq. 8.41)}\\ &= 2: Es(Ta) = 8.1\ eV \quad Es(O) = 1.0\ eV \quad \text{(use eq. 8.42)}\\ &= 3: Es(Ta) = 8.1\ eV \quad Es(O) = 1.0\ eV \quad Es(Ta,0) = 4.55\ eV \quad \text{(use eq. 8.43)}\\ &= 5: Es(Ta) = 8.1\ eV \quad Es(O) = 1.0\ eV \quad Es(Ta,0) = 15.99\ eV \quad \text{(use eq. 8.43)}\end{aligned}$$

The best method is the first model (black line). Note that the covalent binding energy for Ta is unknown, therefore the use of surface binding-model 7 is not possible.

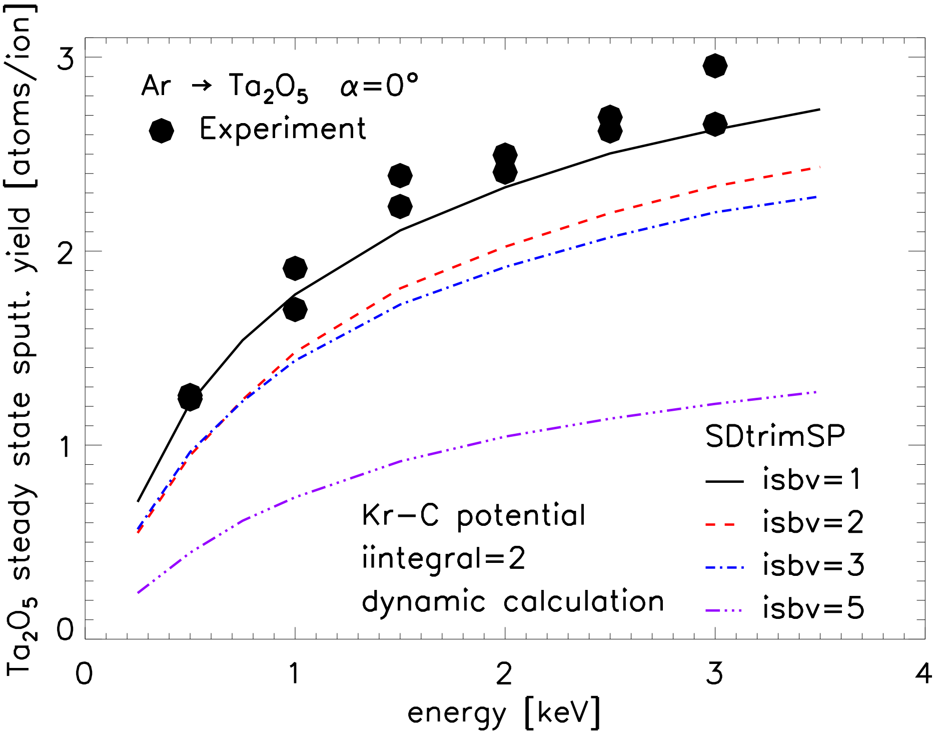

Figure 43: Comparison of sputtered yields calculated with different surface-binding-models (isb=1,2,3,5) and experimental results [40], Ar on Ta_2O_5.

8.4.3. Ar on SiO_2

Another example for the different surface-binding-models is Ar on SiO_2. Fig. 44 shows the comparison of different models with measurements by bombardment of SiO_2 with Ar, [41] - [49]. The measurements have a large spread.

Because the oxygen is a gas the value of surface-binding-energy is not clear. The use of enthalpy of dissociation ΔH^{diss} is only a simple assumption. The reduction of the surface-binding-energy to 1.0 eV provides a good agreement with the measured values also for this example, see also chapter. 8.4.2.
If the target a solid-gas compounds the model 5 (isbv=5) can be used (purple dashed-line). The heat of formation ΔH_f is used for the calculation of Es, [39]. But the agreement of calculated yields with the measured values are not good for this option.
The values for the case Ar on SiO_2 are:

$$\begin{aligned} Es(Si) &= \Delta H_S(Si) = 4.720\ eV \\ \Delta H_f(SiO_2) &= 9.4408\ eV \end{aligned}$$

The surface-binding-energy for the different models (isbv) are:

$$\begin{aligned} \text{isbv} \quad &= 1: Es(Ta) = 4.72\ eV \quad Es(O) = 1.0\ eV \quad \text{(use eq. 8.41)} \\ &= 2: Es(Ta) = 4.72\ eV \quad Es(O) = 1.0\ eV \quad \text{(use eq. 8.42)} \\ &= 3: Es(Ta) = 4.72\ eV \quad Es(O) = 1.0\ eV \quad Es(Ta,0) = 2.86\ eV \quad \text{(use eq. 8.43)} \\ &= 5: Es(Ta) = 4.72\ eV \quad Es(O) = 1.0\ eV \quad Es(Ta,0) = 13.31\ eV \quad \text{(use eq. 8.43)} \\ &= 7: Es(Ta) = 4.72\ eV \quad Es(O) = 1.0\ eV \quad Es(Ta,0) = 5.54\ eV \quad \text{(use eq. 8.43)} \end{aligned}$$

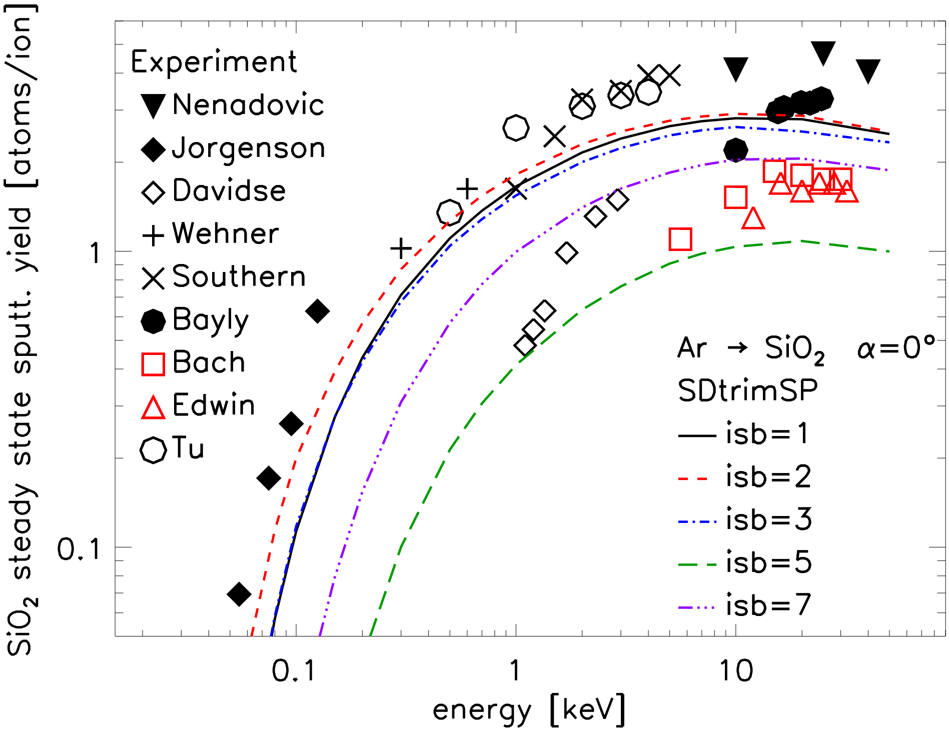

Figure 44: Comparison of sputtered yields calculated with different surface-binding-models (isbv=1,2,3,5,7) and experimental results, [41] - [49], Ar on SiO_2. Note, that the unit of yield by Bach and Edwin (red) are $atoms/ions$ and all other are in $molecules/ion$. These last yields were multiplied with a factor of 3.

The results of this three examples, chapter. 8.4.1, 8.4.2 and 8.4.3, show that the first model (isbv=1) should be used as the default method.

8.5. Sputter yield of low and high fluence Z-projectiles on Cu

The next example is the determination of sputter yields of Cu under impact of 45 keV ions of various atomic number Z for small and high fluences.

Comparison with measurements show that it is better to use for the surface-binding energy of Cu a value a littlebit smaller than ΔH_S.

$$\begin{aligned} Es(Cu) &= 3.20 \\ \Delta H_S(Cu) &= 3.498\ eV \end{aligned} \tag{8.44}$$

Fig. 45 shows the comparison of measured and calculated yields Z on Cu for low fluences. The fluences are $0.5 \cdot 10^{16}$ $atoms/cm^2$, $1.0 \cdot 10^{16}$ $atoms/cm^2$ and near zero in the static mode. The agreement of the calculated values with the experimental data are very good.

Fig. 46 shows the comparison of measured and calculated yields Z on Cu for high fluences. The fluences are $10{\cdot}10^{16}$ $atoms/cm^2$, $100{\cdot}10^{16}$ $atoms/cm^2$ and $500{\cdot}10^{16}$ $atoms/cm^2$.

In most cases for projectiles with $Z > 23$ the steady state has already been reached for a fluence of $10 \cdot 10^{16}$ $atoms/cm^2$. The agreement of calculated yields with the experimental data is qualitatively good. In some cases the exact value is reached. In the case where the yields decreases to zero, e.g. for B and C, the build-up of a solid layer of the collected projectiles was calculated, which prevents any sputtering of the backing material.

Note that the formation of chemical compounds due to O, F, Cl or other elements implantation into materials are not considered. The surface-binding energies were not changed during the calculation. This means that the composition of the surface has no influence on the value of the surface-binding energies. Also, the changes in the structure of the surface layer due to gas agglomeration and bubbling for inert gas were ignored in the model SDTrimSP. Formation of roughnesses has also not been considered.

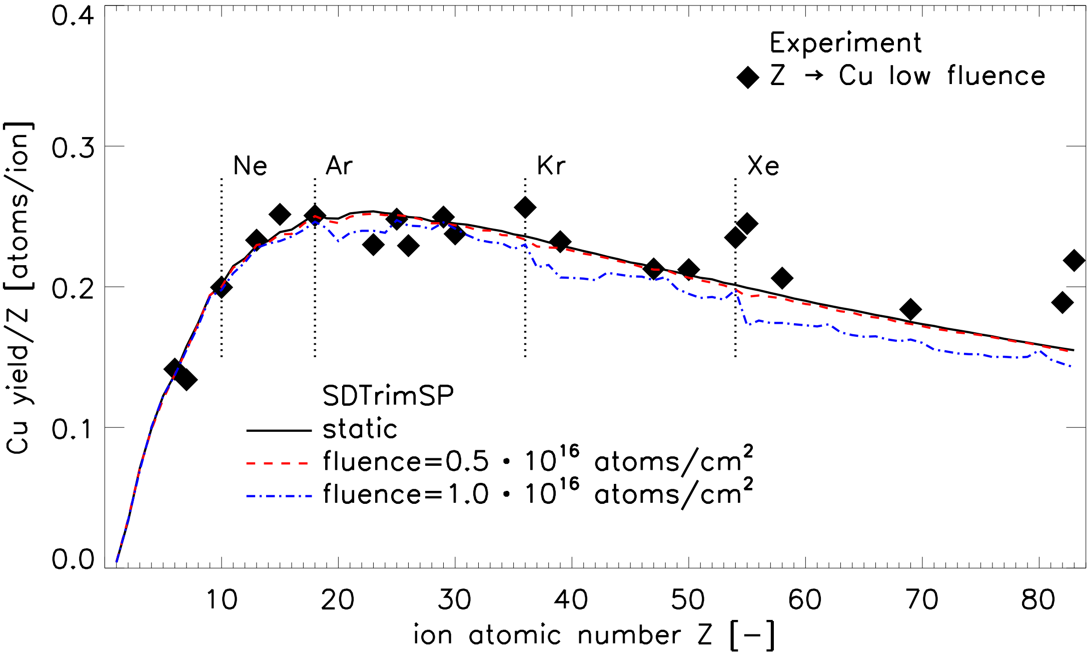

Figure 45: Comparison of calculated sputtered yields low fluence with experimental results [51], Z on Cu.

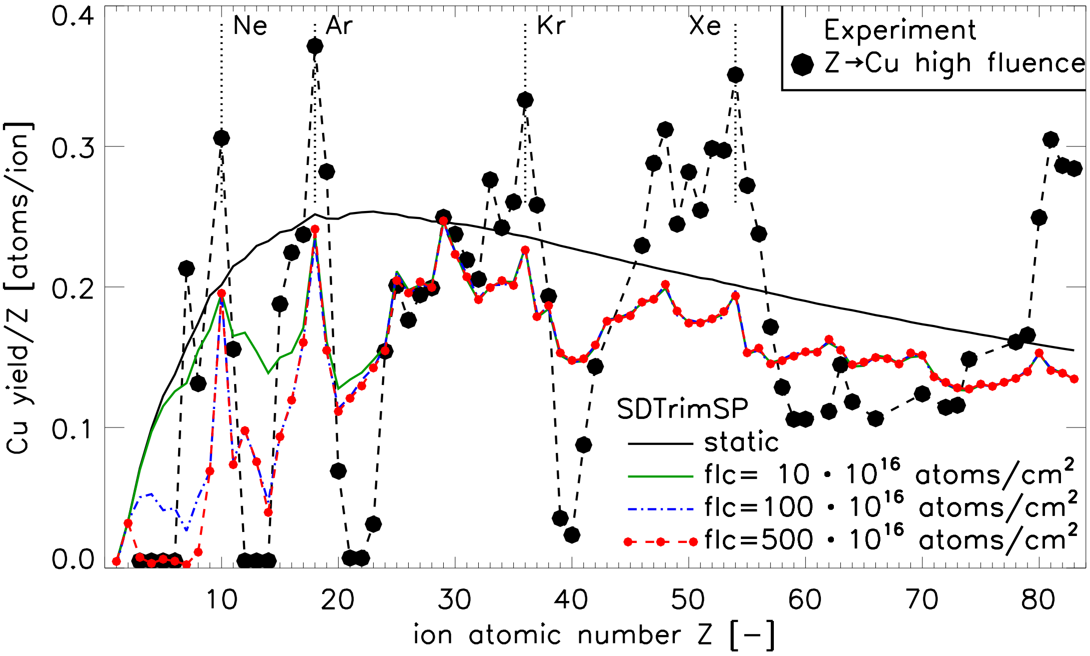

Figure 46: Comparison of calculated sputtered yields high fluence with experimental results [50], Z on Cu. Note Li and Be are calculated for $1000 \cdot 10^{16} atoms/cm^2$

References

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

[2] J. P. Biersack, W. Eckstein, Appl. Phys. A 34 (1984) 73

[3] W. Möller, W. Eckstein, Nucl. Instrum. Meth. B 2 (1984) 814

[4] W. Möller, W. Eckstein, J. P. Biersack, Comput. Phys. Comm. 51 (1988) 355

[5] H. H. Andersen, J. F. Ziegler: In Hydrogen Stopping Powers and Ranges in All Elements, The Stopping and Range of Ions in Matter, Vol.3, ed. by J. F. Ziegler (Pergamon, NewYork, 1977)

[6] J. F. Ziegler: In Helium Stopping Powers and Ranges in All Elements, The Stopping and Range of Ions in Matter, Vol.4, ed. by J. F. Ziegler (Pergamon, NewYork, 1977)

[7] W. D. Wilson, L. G. Haggmark, J. P. Biersack, Phys. Rev. 15 (1977) 2458

[8] J. F. Ziegler, J. P. Biersack, U. Littmark: The Stopping and Range of Ions in Solids, The Stopping and Range of Ions in Matter, Vol.1, ed. by J. F. Ziegler (Pergamon, NewYork, 1985)

[9] G. Molière, Z. Naturforsch. A2 (1947) 133

[10] W. Eckstein, S. Hackel, D. Heinemann, B. Fricke, Z. Phys. D 24 (1992) 171

[11] J. P. Biersack, L. G. Haggmark, Nucl. Instrum. Meth. 174 (1980) 257

[12] K. Mehler, J. Reine Angew. Math. 63 (1864) 152

[13] Z. Kopal, *Numerical Analysis*, Chapman and Hall, London 1961, p. 367 ff.

[14] H. G. Schlager, W. Eckstein, IPP-Report 9/69, Garching 1991

[15] M. T. Robinson: Tables of Classical Scattering Integrals, Oak Ridge Natl. Lab., Oak Ridge, Tennessee (1970)

[16] W. Eckstein, R. Dohmen, Nucl. Instrum. Meth. B 129 (1997) 327

[17] R. Becerra-Acevedo, J. Bohdansky, W. Eckstein and J. Roth, Nucl. Instrum. Meth. B 2 (1984) 631

[18] W. Eckstein, Nucl. Instrum. Meth. B 27 (1987) 78

[19] S. T. Nakagawa, Y. Yamamura, Radiat.Eff.105, 239(1988)

[20] S. T. Nakagawa, Radiation Effects and Defects in Solids, Vol.116 (1991), page 21-28

[21] A. Mutzke, W. Eckstein, Nucl. Instr. and Meth. B 266(2008) 872

[22] K. Wittmaack, Nucl. Instr. and Meth. B 267(2009) 2846-2857

[23] N. Menzel and K. Wittmaack, Nucl. Instr. and Meth. in Phys. Res. B7/8 (1985) 366-370

[24] P. Blank, K. Wittmaack and F. Schulz, Nucl. Instr. and Meth. 132 (1976) 387-392

[25] W. Eckstein, R. Dohmen, A. Mutzke, R. Schneider, Report IPP-Report 12/3, Garching, (2007)

[26] W. Möller, W. Eckstein, Report IPP-Report 9/64, Garching, (1988)

[27] R.A. Zuhr, J. Roth, W. Eckstein, U. von Toussaint, J. Luthin, J. of Nucl. Mater. 290-293 (2001) 162-165

[28] W. Eckstein, C. García-Rosales, J. Roth, and W.Ottenberger, Sputtering Data, IPP-Report 9/82, 1993

[29] J. Roth, J. Bohdansky and W.Ottenberger, Data on low energy, IPP-Report 9/26, 1979

[30] J. F. Ziegler, J. P. Biersack, U. Littmark, *The Stopping and Range of Ions in Solids, The Stopping and Range of Ions in Matter*, Vol.1, Pergamon, New York, 1985

[31] K. Wittmaack, Phys. Rev. B 68, 235211 (2003)

[32] J. Roth, C. Garcia-Rosales, Nuclear Fusion 36 (1996) 1647

[33] C. Hopf, A. von Keudell, W. Jacob, J. Appl. Phys. 94 (2003) 2373-2380

[34] B. V. Mech, A. A. Haasz and J. W. Davis, J. Appl. Phys. 84 (1998) 1655-1669

[35] A. Rai, A. Mutzke and R. Schneider, Nucl. Instr. Meth. B 268 (2010) 2639-2648

[36] J. W. Davis, A. A. Haasz, P. C. Stangeby, J .Nucl. Mater 145-147 (1987) 417-420

[37] W. Eckstein, V.I. Shulga, J. Roth, Nucl. Instr. and Meth. B 153 (1999) 415-421

[38] Y. Kudriavtsev, A. Villegas, A. Godines, R. Asomoza, Appl. Surface Science, 239 Issues 3-4 (2005)

[39] W. Möller, M. Posselt, Report FZR-317, Forschungszentrum Rossendorf, Dresden, (2002)

[40] C. P. Hunt and M. P. Seah, (1983) Surf. Interface Anal. 5 (1983) 199-209

[41] T.Nenadovic, B. Perraillon, Z. Bogdanov,Z. Djordjevi and M. Mili, Nucl. Instrum. Meths. B 48 (1990) 538-543

[42] G. V. Jorgenson and G. K. Wehner, J. Appl. Phys. 36 (1965) 2672-2674

[43] P. D. Davidse and L. I. Maissel, J. Vac. Sci. Technolog. 4 (1967) 33-36

[44] G. K. Wehner, Genaral Mills Report No. 2309, unpublished (1962)

[45] A. L. Southern, W. R. Willis and M. T. Robinson, J. Appl. Phys. 34 (1963) 153

[46] A. R. Bayly and P. D. Townsend, Opt. Laser Technol (1970) 117-121

[47] H. Bach, Nucl. Instrum. Meths. 84 (1970) 4-12

[48] R. P. Edwin, J. Phys. D 6 (1973) 833-841

[49] Y. Y. Tu, T. J. Chuang and H. F. Winters, Phys. Rev. B 23 (1981) 823-835

[50] O. Almèn and G. Bruce, Nucl. Instrum. Meths. II (1961) 279-289

[51] H. H. Andersen and H. Bay, radiation effects 1972, Vol. 13, pp. 67-74

[52] www.srim.org

[53] M.J. Norgett and M.T. Robinson and I.M. Torrens, Nuclear Engeneering and Design 1975, Vol. 33, pp. 50-54

[54] R.E. Stoller, M.B. Toloczko, G.S. Was, A.G. Certain, S. Dwaraknath, F.A. Garner, Nucl. Instr. Meth. B 310, p. 75-80 (2013)

[55] ASTM Standard E521, 2016, ASTM International, West Conshohocken, PA, 2016, DOI: 10.1520/E0521-16

[56] A. Mutzke, R. Schneider, G. Bandelow, SDTrimSP-2D: Simulation of Particles Bombarding on a Two Dimensional Target Version 2.0, IPP-Report 12/11, 2013

[57] R. Arredondoa, M. Oberkofler, T. Schwarz-Selinger, U. von Toussaint, V.V. Burwitza, A. Mutzke, E. Vassallo, M. Pedroni, Nuclear Materials and Energy, 2019 Vol. 18, pp. 72-76

A. Global parameters

parameter	value	description	program
ncpm	8	maximum number of elements	param.F90
nqxm	5000	maximum number of depth intervals	param.F90
pemax	128	maximum number of PEs	work.F90
ntqmax	100000	size of local task queue	default_init.F90

Table 5: Global parameters (set in programs)

B. Input variables in 'tri.inp'

B.1. Necessary input variables

The sequence of the input values in the input file is arbitrary (namelist)

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
	$e0 = ttemp * boltzm$ (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 $atoms/A^2$)
ipot	interaction potential: = 1 : KrC
	= 2 : Moliere
	= 3 : ZBL
	= 4 : Nakagawa-Yamamura
	= 5 : Si-Si
	= 6 : power

Table 6: Necessary input variables (no default values)

variable	description
isbv	surface binding model, determines the composition dependent surface binding energy sbv(ncp,ncp) from the elemental surface binding energies e_surfb(ncp) taken from table1 = 1 : sbv(ip,jp)=e_surfb(jp) for ip=jp, =0 else = 2 : sbv(ip,jp)=e_surfb(jp) for all ip, jp = 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 = 4 : sbv(ip,jp)=f(e_surfb,qu,deltahf) for solid/solid compound = 5 : sbv(ip,jp)=f(e_surfb,qu,deltahf,deltahd) solid/gas compound = 6 : input of given matrix of the surface-binding-energy input-file: mat_surfb.inp = 7 : sbv(ip,jp)=f(electronegativity)
ncp	number of species (projectiles + target species) more than one projectile species is allowed
nh	number of histories (projectiles)
nqx	number of depth intervals of the target (discretization)
qubeam(ncp)	projectile atomic fractions (in incident beam) of ncp species, qubeam > 0. , Note: sum(qubeam(1:ncp))=1 qubeam ≤ 1. for projectiles qubeam = 0. for target atoms
qu(ncp)	initial target atomic fractions of ncp species in case of homogenous initial composition (iq0 = 0)
symbol(ncp)	ncp chemical symbols of elements according to table1 (special symbol: 'H','D','T','He3','He','P_w','P_r', carbon with different density: 'C_a','C_g','C_f','C_d', hybridization state of carbon Sp2,Sp3,Sp3H)
two_comp	symbol of two-component target according to table.compound (e.g. two_comp ='Ta2O5') Note: only selected compounds in table.compound

Table 7: Necessary input variables (no default values) (continued)

B.2. Optional input variables

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
angleinp	'./'	directory of input-file 'angle.inp' (see also: layerinp, tableinp, energyinp)
a_mass(ncp)	table	mass (in amu) of ncp elements; default from table1
a_num_z(ncp)	table	atomic number of ncp elements; default from table1
case_alpha	0	flag for the choice of the angle of incidence = 0 : angle of incidence (degree) counted from the surface normal (azimuthal angle phi = 0°) alpha0 =$-90°$...90° (starting above surface) alpha0 =90°...180° (starting in solid, only static mode) = 1 : random distribution of angles of incidence (only from above surface) (alpha and phi random) alpha0 = 0°..90°, phi = 0°...360° = 2 : cosine distribution of angles of incidence (only from above surface) cos(alpha0) =$\sqrt{0...1}$, phi = 0°...360° = 3 : cosine distribution of angles of incidence cos(alpha0) =$\sqrt{1-(0...1)^2}$, phi = 0°...360° = 4 : input of a given incident angular distribution from input-file: angle.inp = 5 : series of calculations with different angles of incidence (alpha= $(i-1)$·alpha0; i = 1, number_calc) output : output.* dat default set :lmatrices = .false. ltraj_p = .false., ltraj_r = .false. lparticle_r = .false., lparticle_p = .false. case_e0 = 0 (note: all *.dat outputfile from last calculation) = 6 : distribution of alpha and energy input-file: 'ene_ang.inp'

Table 8: Optional input variables with default values

variable	default value	description
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 of projectiles = 3 : temperature (eV) of a Maxwellian energy distribution of projectiles = 5 : series of calculations with different projectile energies e0(1)>0: linear energy= $i \cdot e0$; i = 1, number_calc e0(1)<0: logarithmic energy= $10^{(i-1)} \cdot e0$; i = 1, number_calc output: output.*dat default set: lmatrices = .false. ltraj_p = .false., ltraj_r = .false., lparticle_r = .false., lparticle_p = .false., case_alpha = 0 (note: all *.dat file from last calculation) = 6 : distribution of alpha and energy, input-file: 'ene_ang.inp'
case_layer_thick	0	mixing chema of target = 0 mixing layer with neighbour layer, if thick: 150% or 50% = 1 mixing the whole target to constant layer thick, if one layer thick: 105% or 95% = 2 mixing only the penetration depth and mix with 1/3 method, if one layer thick: 105% or 95%
ca_scre(ncp,ncp)	1.	correction factor for the screening length in the interaction potential (not applicable for KrC and ZBL potentials)
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)

Table 9: Optional input variables with default values (continued)

variable	default value	description
ck_elec(ncp,ncp)	1.	correction factor for the inelastic energy loss; correction factors for hydrogen (below 25 keV) are given in table3
deltahd(ncp)		heat of dissociation (eV) of a molecular target default from table1
deltahf		heat of formation (eV) of a molecular target default from table.compound
diff_koeff1(ncp)	0.0	damage-diffusion-coefficient if loutgas true $[A^4/ion]$ (see also: loutgas)
diff_koeff2(ncp)	0.0	pressure-transport-coefficient if loutgas true $[A^3/ion]$ (see also: loutgas)
dist_nx	60	x-size of the matrix of energy distribution in target
dist_ny	60	y-size of the matrix of energy distribution in target
dist_nz	60	z-size of the matrix of energy distribution in target
dist_delta	2.0	distance between the matrix points of energy distribution in target
dns0(ncp)		atomic density ($atoms/A^3$) of ncp elements; default from table1
dsf	5.	average depth (A) for surface composition
e_bulkb(ncp)	0.	bulk binding energy; if e_bulkb<0. read from table1 e_bulkb is subtracted from the transfer energy
e_cutoff(ncp)		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 selfbombardment)
e_displ(ncp)		displacement energy (eV); default from table1
e_surfb(ncp)		surface binding energy (eV) (heat of sublimation); default from table1
energyinp	'./'	directory of input-file 'energy.inp' (see also: layerinp, tableinp, angleinp)
flux	1.0	flux of incident atoms ($atoms/A^2/s = 10^{20} atoms/m^2/s$)
idrel	1	mode of simulation = 0 : full dynamic calculation (TRIDYN) > 0 : suppression of dynamic relaxation (TRIM), full static calculation < 0 : suppression of dynamic relaxation and cascades static calculation (TRIM) only projectiles (no recoils) are followed

Table 10: Optional input variables with default values (continued)

variable	default value	description
idout	-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 nh/100, 100 fluence steps = 0 : output only after the last fluence step > 0 : output after each idout'th fluence step and last step
iintegral	2	integration method = 0 : MAGIC, only valid for KrC, ZBL, Moliere = 1 : Gauss-Mehler quadrature, ipivot ≥ 8 recommended = 2 : Gauss-Legendre quadrature, ipivot ≤ 16
imcp	2	flag indicating whether (flib)-moments of distributions are calculated = 0 : no moment calculation = 1 : moments of depth distributions for all projectiles (qubeam>0.)
inel0(ncp)	3	inelastic loss model = 1 : Lindhard-Scharff; nessary 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; nessary condition:$E < 25 \cdot Z^{4/3} \cdot M$ (in keV) = 3 : equipartition of 1 and 2 = 4 : high energy hydrogen (H,D,T) (energy > 25 keV) values from 'table3' = 5 : high energy helium (He3,He) (energy > 100 keV) values from 'table4' = 6 : values is calculated for each element use values from 'table6a' and 'table6b'

Table 11: Optional input variables with default values (continued)

variable	default value	description
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	8	number of pivots in the Gauss-Mehler and Gauss-Legendre integration, the minimum number is 4 (larger numbers increase the computing time)
iq0	1	initial composition flag < 0 : initial depth dependent composition taken from file layer.inp $= 0$: initial composition homogeneous, one layer with constant depth intervals
irand	1	random seed
irc0	-1	flag for subthreshold recoil atoms < 0 : subthreshold recoil atoms free ≥ 0 : subthreshold atoms bound, if $E_{start} < E_{displ}$ then replace
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)
i_two_comp	1	method to determine the densities dns0(:) from the compound density in a two-component target (table.compound) =1 : dns0 for the first target species is set equal to the elemental density; necessary if the second element is a gas (e.g. Ta2O5) =2 : dns0 for the second target species is set equal to the elemental density =3 : iterative determination of both dns0(:); recommended if the elemental densities are different

Table 12: Optional input variables with default values (continued)

variable	default value	description
iwc	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	number of ring cylinders for weak simultaneous collisions for recoils
k_start	0	start counter intern
layerinp	’./’	directory of input-file ’layer.inp’ (see also: tableinp, angleinp, energyinp)
lchem_ch	.false.	calculation with chemical erosion H on C(SP2,SP3,SP3H), D on C(SP2,SP3,SP3H)
lenergy_distr	.false.	output of energy distribution in target in E_distr_stop.dat (energy of stop, electric loss and elastic nuclear loss)
lmatrices	.false.	.true. : output of matrices, if idrel /= 0 .false. : no matrix output
lmatout_log_energ	.false.	energy spacing = .false. : linear energy intervals = .true. : logarithmic energy intervals
lmatout_cos_angle	.false.	angular spacing = .false. : angle in degree intervals = .true. : cosine intervals
lmoments	.true.	output of moments for energy distributions (linear and logarithmic) of projectiles and recoils and for range distributions (linear) of projectiles .true. : moments are written .false. : moments are not written
loutgas	.false.	calculation with outgasing diffusion(DDF) and transport (PDF) (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)
lpart_r_ed	.true.	.true. : output of stop recoil information only grater e_displ .false. : output of all stop recoil information
l_pot_thick	.true.	.true. : thickness of surface potential is a funktion of N .false.: thickness of surface potential is a funktion of N,iwc,iwcr

Table 13: Optional input variables with default values (continued)

variable	default value	description
lrestart	.false.	.true. : output of restartfiles after each idout .false. : no restart-files
ltableread	.true.	.true. : read from table1, table2, table3, table4 or table.compound .false. : no table read, a_num_z, a_mass, dns0, e_surfb e_displ have to be given *table*1 : chemical symbol (symbol), nuclear charge (a_num_z), atomic mass (a_mass), mass density, atomic density (dns0), surface binding energy (e_surfb), displacement energy (e_displ), cutoff energy (e_cutoff) *table*2 : chemical symbol, nuclear charge, isotope mass, atomic weight (in amu), natural abundance *table*3 : inelastic stopping coefficients for hydrogen: symbol, nuclear charge, inelastic stopping coefficients a1 to a12 (ch_h), ck *table*4 : inelastic stopping coefficients for helium: symbol, nuclear charge, inelastic stopping coefficients a1 to a9 (ch_he) *table.compound* : symbol of two-component target and physical values
lterm_dif	.false.	.false. : nothing calculate .true. : calculation of thermal diffusion input of given matrix a_0 and e_act input-files: a0_tdiff.inp, eact_tdiff.inp
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 14: Optional input variables with default values (continued)

variable	default value	description
matrix_e_min	0	minimum of lin. energy distribution in matrices
matrix_e_max	max(e0)	maximum of lin. energy distribution in matrices
nm	-1	=-1 : not a molecular target $>$ 1 : number of atoms in a two-component molecule (nm=nm1+nm2)
nm1	-1	number of first atoms in a two-component molecule
nm2	-1	number of second atoms in a two-component molecule
nr_pproj	10	number of projectiles between two target updates (idrel = 0)
numb_hist	20	number of traced trajectories of projectiles and recoils
number_calc	1	number of calculations if a series of calculations is carried out (case_e0 = 5 or case_alpha = 5)

nx_mat	1000	intervall of depth origin
		nx_mat= 100 intervall depth origin: input target '
		nx_mat=1000 intervall depth origin: 1 A '
		nx_mat=1001 intervall depth origin: 0.5 A '

qu_int	.false.	linear interpolation of atomic fractions between the depth intervals
qumax(ncp)	1.	maximum atomic fractions in the target for ncp species, if idrel=0
rhom		atomic density of a two-component target; default from table.compound $[g/cm^3]$
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 $= 3 \cdot \|e0\|$, only if case_e0=2,3 (Maxwellian distribution, plasma)
tableinp	'../../tables'	directory of input-file for tables (see also: layerinp, angleinp, energyinp)
text		comment in NAMELIST
ttarget		total target thickness in Angstrom (A)
ttemp	300.	target temperature, only for thermical diffusion and high temperatures, it reduces the surface binding energy according to a Maxwellian energy distribution
x0(ncp)	0.	starting position of projectile $\leq$ 0. : outside the surface at $x = xc = -su$ $>$ 0. : inside the solid

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

C. Output format

C.1. Output format of energy distribution in the target

The option *lenergy_distr* enables the calculation of the distribution of the energy input in the target. The total input energy is divided into the contributution of stopped atoms, inelastic energy loss and elastic energy loss.

option: lenergy_distr = .true.

parameter: dist_nx
dist_ny
dist_nz
dist_delta

output: E_distr_all.dat (output of total input energy)
E_distr_inel.dat (output of inelastic energy loss)
E_distr_nucl.dat (output of elastic energy loss)
E_distr_stop.dat (output of energy from implanted particle)

The default values are:

variable	number	interval [Å]	distant [Å]	description
dist_nx	60	2	0 to 120	depth
dist_ny	60	2	-61 to 59	width
dist_nz	60	2	-61 to 59	length

Table 16: Default values for the option lenergy_distr

C.2. Output format of the depth of origin and penetration depth

In the static mode the output of the depth of origin and penetration depth are possible.

option: lmatrices =.true.

optional option: lmatout_log_energ=.true.

output: morigin_ex_bs.dat ...depth of origin of backsputtered atoms dependent on energy
morigin_ex_ts.dat ...depth of origin of transmitted sputtered atoms dependent energy
mpe_ex_p.dat ...maximum of penetration of backscattered atoms
mepb.dat ...path-length of backscattered atoms dependent energy
mept.dat ...path-length of backscattered atoms dependent energy

C.3. Output format of energy- and angle-distribution of scattered and sputtered atoms

C.3.1. Output-matrix-file

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

option: lmatrices =.true.

optional option: lmatout_cos_angle=.true.

output:	meagb_p.dat	...output of backscattered particles
	meagb_s.dat	...output of all backsputtered particles
	meagt_p.dat	...output of all transmitted scattered particles
	meagt_s.dat	...output of all transmitted sputtered particles

C.3.2. 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)	azimut(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)	azimut(lin)	number of particles per solid angle
matrixc_eg..	energy(lin)	azimut(lin)	number of particles
matrixc_ee..	polar(cos)	azimut(lin)	energy
matrixclea..	energy(log)	polar(cos)	number of particles
matrixcleg..	energy(log)	azimut	number of particles

file extension:

..b_p..	backscattered 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. backscattered-projectile dependent on polar- and azimut-angles)

C.4. Output format of depth_proj.dat und depth_recoil.dat

(see also subsection 7.4)

depth_proj.dat stores depth profiles of projectile target interaction information
Columns:

Column name	Description
DEPTH/LENGTH	Center of layer or particle trajectory length
STOPS	Number of stopped projectiles in layer
PATHLENGTH	Number of particles with a trajectory of this length (sampled on same scale as target thickness → If pathlength > ttarget poor sampling)
NUCL.LOSS	Energy deposited at this depth by elastic collisions
ELECT.LOSS	Energy deposited at this depth by inelastic interaction with electrons
NUC.LOSS < ED	Number of elastic collisions where transferred energy < Displacement energy
NUC.LOSS > ED	Number of elastic collisions where transferred energy > Displacement energy
FRENKEL P.	Kinchen Pease based DPA
NUM.COLL.	Not used
NRT-DPA	Norgett-Robinson-Torrens [53] model based DPA

depth_recoil.dat stores depth profiles of recoil cascade information
Columns:

Column name	Description
DEPTH/LENGTH	Center of layer particle trajectory length
STOPS	Number of stopped projectiles in layer
PATHLENGTH	Number of particles with a trajectory of this length (sampled on same scale as target thickness → If pathlength > ttarget poor sampling)
NUCL.LOSS	Energy deposited at this depth by elastic collisions
ELECT.LOSS	Energy deposited at this depth by inelastic interaction with electrons
NUC.LOSS < ED	Number of elastic collisions where transferred energy < Displacement energy
NUC.LOSS > ED	Number of elastic collisions where transferred energy > Displacement energy
VACANCIES	Number of displacements where a vacancy is left behind
DISPLCNTR	Number of collisions where the transferred energy > Displacement energy
REPLACEMENT	Number of displacements where the projectile replaces the recoil

D. Inputfile 'layer.inp'

number depth	thick-	composition of target (2:ncp)
interval	ness	qu_2
1	0.10000E+02	0.94061E+00
1	0.10000E+02	0.89911E+00
1	0.10000E+02	0.88149E+00
1	0.10000E+02	0.86192E+00
3	0.10000E+02	0.85698E+00
3	0.10000E+02	0.84040E+00
1	0.10000E+02	0.83929E+00
1	0.10000E+02	0.82595E+00
1	0.10000E+02	0.82483E+00
1	0.10000E+02	0.81702E+00
1	0.10000E+02	0.81551E+00
1	0.10000E+02	0.81049E+00
100	0.10000E+02	0.80643E+00
0	0	0

Table 17: Inputfile 'layer.inp' for two components (ncp=2), $qu\_1 = 1 - qu\_2$

number of	thick-	composition of	target (2...ncp)
layer	ness	qu_2	qu_3
200	5.0	0.3	0.7
300	5.0	0.7	0.3
0	0	0	0

Table 18: Inputfile 'layer.inp' for three components (ncp=3), $qu\_1 = 1 - sum(qu(2:ncp)) = 0.0$

E. Inputfiles 'energy.inp' and angle.inp

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

Table 19: Inputfile 'energy.inp'

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

Table 20: Inputfile 'angle.inp'

F. Inputfiles 'ene_ang.inp'

0.000000			: help	value		
0.000000			: help	value		
9			: columns	Angles		
10.0000	80.00000	10.00000	: Min Max	dAngle	[degrees]	
13			: rows	Energy		
10.0000	140.0000	10.00000	: Min Max	dE	[eV]	
2.39560	2.728417	9.982015	4.214628	1.330935	4.436451	3.327338
1.55830	5.512290	1.098021	2.994604	2.218225	1.109112	0.000000
3.56800	5.079736	3.438249	4.436451	2.218225	1.109112	0.000000
3.32730	2.218225	1.109112	1.109112	1.109112	0.000000	0.000000
0.00000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.00000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.00000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.00000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.00000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.00000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.00000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.00000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.00000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

Table 21: Inputfile 'ene_ang.inp'

G. Example of Inputfile ’tri.inp’

Inputfile ’tri.inp’ of first static example He − > Ni

```
2 keV He − > Ni
&TRI_INP

    text=’—elements—’
       ncp = 2
       flc = 10.0
       nh = 10
       idout = 10
       nr_pproj = 1

       idrel = 1
       isbv = 1
       ipot = 1

    text=’—beam—’
       qubeam = 1.000, 0.000
       qumax = 0.000, 1.000
       case_e0 = 0
       e0 = 2000, 0.00
       case_alpha = 0
       alpha0 = 0.000, 0.000

    text=’—target—’
       ttarget = 5000E+0
       nqx = 500
       qu = 0.0 , 1.0
       e_cutoff= 1.0 ,1.0

       ltraj_p = .true.
       ltraj_r = .true.
       numb_hist = 1
       ioutput_hist = 1, 1, 0, 1, 1, 0
       lparticle_r = .true,
       lparticle_p = .true.
       ioutput_part = 100, 100, 0, 100, 100, 0
/
```

Inputfile ’tri.inp’ of second static example Ar − > Ni

```
1 keV Ar − > Ni
&TRI_INP

   text=’—elements—’
       ncp = 2
       symbol = ”Ar”, ”Ni”

       flc = 10.000E+0

       nh = 10000000
       idout = 100000
       nr_pproj = 10
       idrel = 1
       ipot = 1
       isbv = 1

   text=’—beam—’
       qubeam = 1.000, 0.000
       qumax = 0.000, 1.000
       case_e0 = 0
       e0 = 1000, 0.00
       case_alpha = 0
       alpha0 = 60.000, 0.000

   text=’—target—’
       ttarget = 5000E+0
       nqx = 500,
       qu = 0.0 , 1.0

       lmatrices = .true.
/
```

Inputfile ’tri.inp’ of first dynamic example Ni − > WC

```
10 keV Ni − > WC
&TRI_INP

   text=’—elements—’
      ncp = 3
      symbol = ”Ni”,”W”, ”C_g”
      flc = 15.00
      nh = 45000
      idout = 500
      nr_pproj = 64

   text=’—beam—’
      qubeam = 1.000, 0.000, 0.000
      qumax = 1.000, 1.000, 1.000
      case_e0 = 0
      e0 = 10000, 0.00
      case_alpha = 0
      alpha0 = 0.0 , 0.000, 0.000

      ipot = 1
      isbv = 4
      inel0 = 3 ,3 ,3

   text=’—target—’
      nm=2
      two_comp=’WC’
      idrel = 0
      ttarget = 500
      nqx = 100
      qu = 0.0, 0.5, 0.5
/
```

Inputfile ’tri.inp’ of second dynamic example W − > C

```
5 keV W − > C
&TRI_INP

  text=’—elements—’
     ncp = 2
     symbol = ”W”, ”C_g”

     flc = 50.00
     nh = 1000000
     idout = 500
     nr_pproj = 64

  text=’—beam—’
     qubeam = 1.000, 0.000
     qumax = 1.000, 1.000
     case_e0 = 0
     e0 = 5000, 0.00
     case_alpha = 0
     alpha0 = 0.00, 0.00

     ipot = 1
     isbv = 1
  text=’—target—’
     idrel = 0
     ttarget = 1000
     nqx = 100
     qu = 0.0, 1.0
     qu_int = .true.
     case_layer_thick = 2
     e_cutoff= 1.0, 1.0
/
```

Inputfile 'tri.inp' of third dynamic example Ar − > Si Ta

```
3 keV Ar − > Si Ta
&TRI_INP
   text='—elements—'
       ncp = 3
       symbol = ”Ar”, ”Si”, ”Ta”
       flc = 50
       nh = 50000
       idout = 500
       nr_pproj = 32
       ipot = 1
       isbv = 3
   text='—beam—'
       qubeam = 1.000, 0.000, 0.000
       qumax = 0.000, 1.000, 1.000
       case_e0 = 0
       e0 = 3000, 0.00
       case_alpha = 0
       alpha0 = 0.000, 0.000 ,0.000
   text='—target—'
       idrel = 0
       case_layer_thick = 2
       ttarget = 1815
       nqx = 363
       qu = 0.0, 0.5, 0.5
       iq0 = -1
/
```

Inputfile 'layer.inp' of third dynamic example Ar − > Si Ta

number of	thick-	target composition 2...ncp		name of layer
layer	ness	qu_2	qu_3	
40	5.00	1.0000	0.0000	Si 1
15	5.00	0.0000	1.0000	Ta 1
21	5.00	1.0000	0.0000	Si 2
15	5.00	0.0000	1.0000	Ta 2
21	5.00	1.0000	0.0000	Si 3
15	5.00	0.0000	1.0000	Ta 3
21	5.00	1.0000	0.0000	Si 4
15	5.00	0.0000	1.0000	Ta 4
200	5.00	1.0000	0.0000	Si 5
0	0	0	0	end

Inputfile 'tri.inp' of sputtering with noble gas ions Xe − > Si

```
10 keV Xe − > Si
&TRI_INP
   text='—elemements—'
       ncp = 2
       symbol = ”Xe”, ”Si”

   text='—beam—'
       case_e0 = 0
       e0 = 10000, 0.00
       qubeam = 1.00, 0.000
       case_alpha=0
       alpha0 = 0.0 , 0.000

   text='—control—'
       flc = 10
       nh = 10000
       nr_pproj = 80
       idout = 1000
       idrel = 0
       ipot = 1

   text='—target—'
       ttarget = 5000
       nqx = 500
       isbv = 1
       inel0 = 3 ,3
       qu = 0.0, 1.00
       qumax = 1.00, 1.000
       e_cutoff= 1.0, 1.0
       qu_int= .true.
       case_layer_thick=2
       loutgas = .true.
       diff_koeff1 = 1.60e06 , 0.0
       diff_koeff2 = 70 , 0.0
/
```

Inputfile 'tri.inp' of sputtering with chemical erosion H − > C

```
200 eV H − > C
&TRI_INP
   text='—elements—'
       ncp = 4
       symbol =”H”,”Sp2”,”Sp3”,”Sp3H”

       flc = 30
       nh = 30000
       idout = 1000
       nr_pproj = 5000

       idrel = 0
       isbv = 1
       ipot = 1

    text='—beam—'
       qubeam = 1.00 , 0.00 , 0.00, 0.00
       case_e0=0
       e0 = 200 ,0.00, 0.00, 0.00
       case_alpha=0
       alpha0 = 0.00, 0.00, 0.00, 0.00

    text='—target—'
       ttarget = 5000
       nqx = 1000
       qumax = 1.0, 1.00, 1.00, 1.00
       qu = 0.0, 1.00, 0.00, 0.00
       ttemp = 800
       irc0 = 1
       e_surfb = 1.00, 4.00, 4.00, 4.00
       e_cutoff= 1.0, 1.0, 1.0, 1.0
       inel0 = 4, 3, 3, 3

       qu_int = .true.
       case_layer_thick=2
       loutgas = .true.
       diff_koeff1 = 1.00E06, 0.0, 0.0, 0.0
       diff_koeff2 = 100, 0.0, 0.0, 0.0
       lchem_ch = true
       flux = 1.00
/
```

Inputfile ’tri.inp’ of sputtering with thermal diffusion C − > WC

```
1 eV C − > CW with diffusion
&TRI_INP
   text=’—elements—’
       ncp = 2
       symbol =”C_g”,”W”

       flc = .1
       nh = 1000
       idout = 10
       nr_pproj = 8

   text=’—beam—’
       qubeam = 1.00 , 0.00
       qumax = 1.0, 1.00
       case_e0=0
       e0 = 1.0, 0.00
       case_alpha=0
       alpha0 = 0.00, 0.00

       ipot = 1
       isbv = 1
       inel0 = 3, 3

    text=’—target—’
       idrel = 0
       ttarget = 5000
       nqx = 1000
       qu = 0.0, 1.00
       iq0 = 1
       ttemp = 873
       irc0 = 1
       e_cutoff= 1.0, 1.0
       qu_int = .true.
       case_layer_thick=0
       lterm_dif = .true.
       flux = 0.0001
/
```

Inputfile 'a0_tdiff.inp' of example C − > WC

```
a0 [m**2/s]
2   number of elements
6   C Z-symbol for check of input
74   W Z-symbol for check of input

         C          W
 8.91e-6        0.0    a0 for C
       0.0   8.91e-6   a0 for W
```

Inputfile 'eact_tdiff.inp' of example C − > WC

```
a_act [eV]
2   number of elements
6   C Z-symbol for check of input
74   W Z-symbol for check of input

    C      W
 2.32   0.00    a_act for C
 0.00   2.32   a_act for W
```