

IMSIL 2021-09

G. Hobler

TU Wien, AUSTRIA

Release Version
(printed November 2, 2021)

Contents

1	Introduction	5
2	Ion and Target Specification	9
2.1	Ion Beam	10
2.2	Material	12
2.3	Geometry	13
3	Physical Models	15
3.1	Selection of Collision Partners	15
3.2	Interatomic Potential	17
3.3	Implantation Damage	18
3.4	Backscattering, Transmission, and Sputtering	21
3.5	Electronic Stopping	22
3.6	Rutherford Backscattering / Channeling	24
4	Files	27
4.1	Input Files	28
4.2	The Output File OUT	31
4.3	1-D Histogram Files	32
4.4	2-D Histogram Files	37
4.5	3-D Histogram Files	39
4.6	Miscellaneous Output Files	40
5	Input Records	43
5.1	&SETUP	45

5.2	&GEOMETRY	53
5.3	&ATOMS	60
5.4	&IONS	64
5.5	&CRYSTAL	71
5.6	&MATERIAL	74
5.7	&DAMAGE	78
5.8	&SNPAR	87
5.9	&SEPAR	95
5.10	&OUTPUT	101
6	Auxiliary Scripts	123
6.1	Scripts for Running IMSIL	123
6.2	Scripts for Reading and Plotting Results	126
7	Examples	129
7.1	Ion Implantation into a Flat Target	129
7.2	Ion Implantation into a Trench	130
7.3	Ion Implantation into a Multilayer Structure	132
7.4	Spot Burn	133
7.5	Static Sputtering	134
7.6	Dynamic Sputtering	135
7.7	RBS	137

Chapter 1

Introduction

Ion implantation has traditionally been the most important doping technique in semiconductor technology. One-dimensional doping profiles of the most common ion species implanted into (100) Si at 7° tilt angle have accurately and extensively been measured and parameterized. On the other hand, physics-based simulations may be invaluable if less common implant parameters are used (such as large tilt angles or different materials), if the dopant profiles are difficult to measure (such as at ultra-low energies or in two dimensions), or if information on implant damage is desired.

Another related phenomenon is sputtering. Sputtering can occur unintentionally during high-dose ion implantation or intentionally to remove material, possibly in a local manner using a focused ion beam. Other related phenomena occur in nuclear technology such as defect creation in fission and plasma-wall interaction in fusion technology. In all these situations the binary collision (Monte Carlo) IMplant and Sputter sImuLator IMSIL¹ may be the tool of choice.

This manual describes IMSIL as of November 2, 2021.

IMSIL allows targets composed of a crystalline region with a lattice structure that can be described by a cubic unit cell, and an arbitrary number regions of arbitrary amorphous material [1]. The target may either be a 1-D layered structure defined by the interface positions of the materials [2, 3], or a 2-D structure composed of regions defined by arbitrary polygons [4, 5]. For 1-D targets, geometry and material changes may be simulated in a dynamic manner. For all target geometries, damage produced by the ions in the crystalline layer may dynamically be taken into account for the simulation of subsequent ion trajectories [6], which can have a profound effect on the distribution of the implanted ions due to the suppression of the channeling effect. The motion of recoils may be followed in order to obtain a more accurate distribution of the damage [7], to obtain the distribution of atoms recoiled from one target region into another [8], and must be followed to simulate sputtering. The ions may either be atoms (B, P, As, ...) or molecules composed of an

¹earlier, “IMSIL” was considered an abbreviation of IMplant simulator for SILicon technology. The new interpretation reflects the evolution of IMSIL, covering sputtering as well as implantation.

arbitrary number of atoms (BF_2 , ...).

The models implemented in IMSIL have been carefully calibrated against experimental data for conditions commonly found in semiconductor industry. In particular, electronic stopping is well described in silicon both under random and channeling conditions for all implant species.² Damage generation in silicon has been calibrated for B, BF_2 , Si, P, and As, and can be interpolated for any ion species. In order to deal with special conditions such as different materials, target temperatures different from room temperature, or high dose rates, the user is given access to all model parameters via the input file.

IMSIL was first released in 1993. Until 2003, it was mainly developed as an implant simulator. Apart from the carefully calibrated models for silicon technology [1, 9, 10, 11], another unique feature of IMSIL is its physics-based model of extended free flight paths in amorphous targets [12], which promote computational efficiency at high ion energies.

Two major extensions of IMSIL 2005 were the possibility to calculate RBS/C energy and depth spectra [13], and to use a deterministic model of damage as an alternative to the standard statistic model [14, 15]. The deterministic defect model was used for coupling IMSIL with kinetic Monte Carlo simulations. It has been disabled, because it has not been used since then.

The major extension of IMSIL 2007 was the capability to use crystal structures different from diamond, namely wurtzite, zincblende, and the 4H and 6H polytypes of SiC [16]. More recently the possibility to read in arbitrary orthogonal unit cells and the fcc, bcc, and sc lattice were implemented [17]. Also, a polycrystalline mode has been implemented [17] where the crystal lattice is randomly rotated before each ion impact.

Starting 2008, various extensions for more accurate sputtering simulations have been implemented. These include the implementation of a surface binding energy (planar, isotropic and intermediate models) [18, 19] and of a surface roughness model that provides more realistic sputter yields for glancing incidence of the ions [20]. The introduction of “virtual recoils” [21] promotes computational efficiency by following recoils down to the surface binding energy only near the surface, so that both sputtering and displacement statistics are correctly obtained, while unnecessary trajectory simulations down to the surface binding energy in the bulk are avoided. Targets with rotational and spherical symmetry have been implemented in 2016 [22, 23]. Arbitrary 3-D geometries still wait for implementation, although the geometry-related code has been written with the extension to 3-D in mind. Some activities on 2-D dynamic target relaxation have been started [21, 24], and some traces can be found in the code. It is planned to separate this feature from IMSIL into a separate code with some coupling to IMSIL.

As a beam model, traditionally a punctiform beam and a homogeneous windowed beam had been available with an optional simple beam divergence model (isotropic up to a maximum angle). The beam divergence model has recently (2021) been extended to include superpositions of Gaussian distributions with different standard deviation in x and

²Nuclear stopping is generally considered accurate under most conditions anyway. This should be taken for granted for any binary collision simulator.

y direction for both the position and the direction of the incident ions. The former allows to describe quite general focused ion beams, while the latter is relevant to channeling if the beam is close to a low-index crystallographic direction.

IMSIL provides a multitude of options for writing spatial, angular, and energy distributions to output files (“histogram files”). These include the 1-D, 2-D, and 3-D spatial distributions of stopped ions, vacancies, and recoils as well as nuclear and electronic energy deposition. In addition, 1-D and 2-D angular and 1-D energy distributions of backscattered or transmitted ions and recoils can be written to files. Also, ion and recoil trajectories or just their initial and/or final states can be saved to a file.

The remainder of this manual is organized as follows: After an introduction to the options for the specification of the ions and the target in Chapter 2, the physical models implemented in IMSIL are described in Chapter 3. The formats of the input and output files are given in Chapter 4. Chapter 5 describes all the parameters that may be specified in the input file. After a description of Python scripts that facilitate the running of IMSIL and the plotting of results in Chapter 6, typical examples of input files are presented in Chapter 7.

Chapter 2

Ion and Target Specification

IMSIL simulates the interaction of an ion beam with a target. Throughout this manual, **ion** means the particle that is given initial conditions by the user. We use this term irrespective of the atom's actual charge state.¹ The ions may be single atoms or molecules.

Beam denotes the properties of a collection of ions. The term is used here in a broader sense than in common parlance. During conventional ion implantation or sputtering, ions are part of a “beam”. During single-ion implantation, the “beam” consists of only one ion. In this case, when many ions are simulated, the results can be interpreted as probabilities and probability density functions. Atoms starting from inside the target, e.g., as recoils in nuclear reactions, are also considered a “beam” in the context of this manual.

The target consists of one or several **regions** with homogeneous material properties. A crystalline region may be modified during a simulation due to damage formation. While this kind of simulation can be considered **dynamic**, we reserve the word here for sputtering simulations where the geometry and composition of the target changes as a function of time. In this case, the target is defined by **cells** rather than regions. Cells typically have the size of histogram bins in order to resolve the spatial and temporal variations of the target composition as the simulation proceeds. In the code, regions and cells are considered “domains”.

While “ion” refers to a particle that receives initial conditions by specifications in the input file, we use the terms **projectile** and **recoil** when we talk about collisions: The projectile is the energetic atom that impinges on another atom, the recoil, which initially is at rest and receives energy in the collision. We use the term “recoil” irrespective of whether the atom permanently leaves its original site.

¹Ions “forget” their initial charge state after a short distance travelled inside the target.

2.1 Ion Beam

The **ion species** is defined by the chemical name of an atom (e.g., He or B) or of a molecule (e.g., BF₂ for a boron difluoride molecule). In case of a molecule, the **ion energy** is considered the kinetic energy of the molecule, and the individual atom energies are derived based on the atom masses. Each atom of the molecule is implanted one after another. For instance, if the ion is BF₂, then one B atom is followed by two F atoms. Currently, the initial conditions of each molecule atom are treated as independent from each other, i.e., any property defined by a statistical distribution uses different random numbers for each molecule atom. This approximation is not expected to make a difference in most if not all cases. It is also possible to specify the **mass** of the atoms in order to distinguish isotopes.

The **starting points** of the ion trajectories are determined in several steps. First, a reference point is defined. Its coordinates are chosen uniformly distributed in an axis-parallel brick. Degenerate forms of the brick, a rectangle, a line, or a single position are also possible. If a beam profile with a nonzero full-width-at-half-maximum is specified, this reference point is moved in the plane perpendicular to the beam direction by a Gaussian distributed displacement. If a crystalline region is specified and an external beam is chosen, the reference point is additionally moved by a vector between a corner of the crystal unit cell to a random point inside the cell. If at least one crystalline region is specified and the ions are chosen to start inside the target, the reference point is additionally moved to the nearest lattice site. Finally, in case of an external beam and if the reference point is inside the target, the starting point is obtained by projecting the reference point to the surface. This is done along the ion direction to a position which is a distance equal to the maximum impact parameter outside the target, where the ion, when coming from infinity along a straight line, first hits the target. Otherwise, the ion starts at the reference point.

The **initial direction** of the ions is defined by a reference direction and, optionally, an additional rotation due to beam divergence. The reference direction is defined by a tilt and a rotation angle relative to the target coordinate system, see Fig. 2.1. The tilt and rotation angles may be specified, or the rotation angle may be chosen randomly for each ion. Without beam divergence the initial directions of all ions coincide with the reference direction \hat{z}' (cf. Fig. 2.1). Considering a beam divergence means to randomly rotate the ion's direction of motion on the unit sphere from the beam direction in such a way that the probability per solid angle follows a given distribution. Two beam divergence models may be chosen from: isotropic within a maximum rotation angle and Gaussian with given standard deviations in \hat{x}' and \hat{y}' direction.

More refined beam profiles or divergence distributions may be defined by superposing partial beams using the **BEAM** index variable. An index variable used on an input record specifies that the parameters on this record apply to a particular object only, in this case to a partial beam. Each partial beam may have a different beam profile and/or beam divergence. Ions are then chosen with probabilities proportional to a specified beam

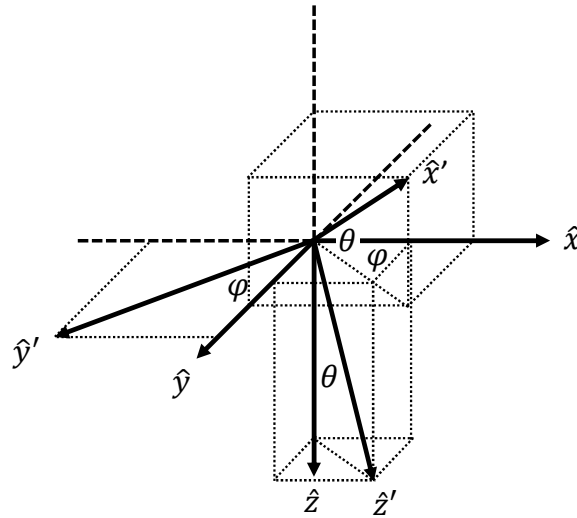


Figure 2.1: Definition of the beam coordinate system $(\hat{x}', \hat{y}', \hat{z}')$ and the target coordinate system $(\hat{x}, \hat{y}, \hat{z})$ by tilt angle θ and rotation angle φ .

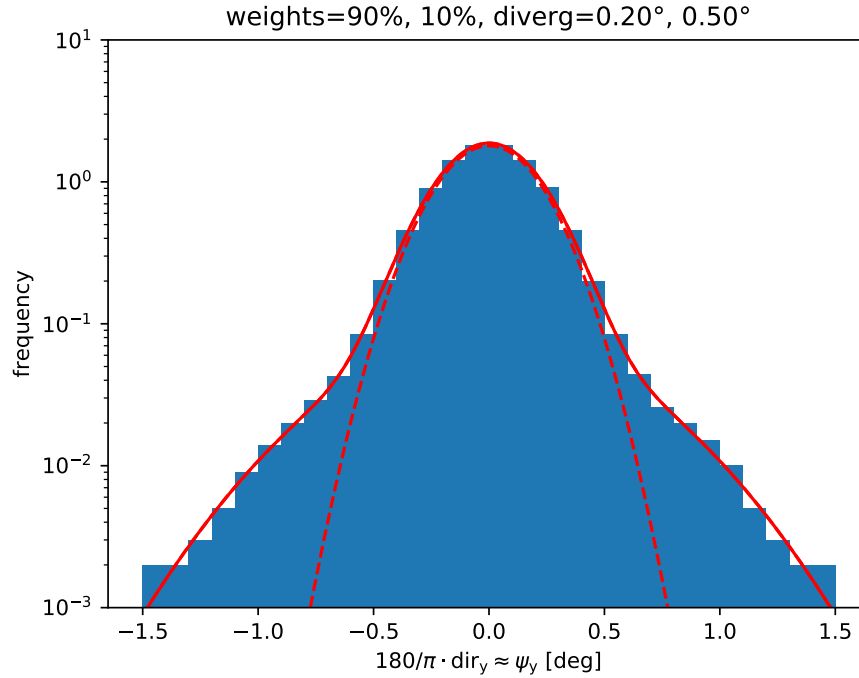


Figure 2.2: Beam profile composed of two Gaussian functions (red line) and histogram of angles ψ_y generated by IMSIL.

weight from each of the partial beams. An example for the superposition of two Gaussian beam divergence distributions is shown in Fig. 2.2. Below is an excerpt from an IMSIL input file that specifies the above beam profile:

```
&IONS NAME='B' ENERGY=5000 DOSE=1e13 TILT=0 MODDIV=2 NION=10000 /
&IONS BEAM=1 WEIGHT=0.9 DIVERG=0.1,0.2 /
&IONS BEAM=2 WEIGHT=0.1 DIVERG=0.2,0.5 /
```

Note that parameters that are specified on the first line, which does not have the BEAM index parameter, apply to both beams.

The **ion dose** may either be specified in ions per area in the x - y plane (areal dose, units cm^{-2}), ions per length in y direction (line dose, units cm^{-1}), or ion count. The most natural choice among these options depends on the dimensionality of the spatial histograms that are computed: areal dose for 1-D, line dose for 2-D, and number doses for 3-D histograms. However, other choices are possible, see Sections 4.3, 4.4, and 4.5.

parameter	IMSIL name	to be specified in record
internal starting point flag	LIINIT	&IONS
ion name	NAME	&IONS
energy	ENERGY	&IONS
atom mass	MASS	&ATOMS
reference starting point	XINIT	&IONS
	YINIT	&IONS
	ZINIT	&IONS
beam FWHM	FWHM	&IONS
tilt angle	TILT	&IONS
rotation angle	ROTATE	&IONS
random rotation flag	RANROT	&IONS
beam divergence model	MODDIV	&IONS
beam divergence	DIVERG	&IONS
beam index variable	BEAM	&IONS
beam weight	WEIGHT	&IONS
dose	DOSE	&IONS
dose units	DOSEUNITS	&IONS
number of simulated ions	NION	&IONS

2.2 Material

For **static simulations**, the target is composed of **regions**. Each region is described by a material and a geometry. The **material** is primarily defined by its name, which is a chemical formula similar to the ion name. The **density** of a region is set to the weighted densities of the elementary materials by default, or can be specified by the user. Note that for compound targets, the weighted densities are often inaccurate, so their densities should be specified in the input file.

Another material property is its **crystallinity**. If the material is crystalline, the crystal structure is defined in the **CRYSTAL** record. In addition, the crystallographic directions of the surface normal and of a surface vector may be specified. Alternatively, the material may be specified as polycrystalline, in which case a new random orientation of the crystal is chosen for each ion.

A further material property is the RMS **thermal vibration** amplitude of the atoms, which may either be calculated from the Debye theory using the target and Debye temperature, or it may be specified directly. Thermal vibrations are only relevant in crystalline targets.

For **dynamic simulations**, the **&MATERIAL** record can be used to specify the initial material name and density. These specifications are converted into atom densities, which are used to initialize the cell contents. The cell densities are then allowed to change during the simulation. Since the other material properties depend on the target composition, they can only be interpolated from the elemental material properties. Crystalline targets are not allowed in dynamic simulations.

parameter	IMSIL name	to be specified in record
number of regions	NR	&SETUP
material name	NAME	&MATERIAL
elementary material density	DENSITY	&ATOMS
material density	DENSITY	&MATERIAL
crystallinity flag	XTAL	&MATERIAL
polycrystallinity flag	LPOLY	&MATERIAL
surface normal	WAFER	&MATERIAL
surface vector	VSURF	&MATERIAL
target temperature	TEMP	&SETUP
Debye temperature	TDEBYE	&MATERIAL
RMS thermal vibration amplitude	VIB	&MATERIAL

2.3 Geometry

IMSIL currently supports 1-D, 2-D, and rotational symmetric 3-D geometries for static simulations, and 1-D geometries for dynamic simulations.

1-D geometries describe layered targets with surfaces and interfaces parallel to the x - y plane. They are therefore defined by an array of z coordinates which correspond to the position of the top surface, any interfaces between layers (regions), and the bottom surface. Semi-infinite and infinite targets may be approximated by large enough values of the surface coordinates.

1-D geometries combined with **cylindrical or spherical symmetry** describe targets composed of cylindrical or spherical shells. The surface and interface positions then refer to the distance from the symmetry line or center. The “top” surface is the inner surface,

and the “bottom” surface the outer surface. If a hollow body is not desired, the position of the inner surface has to be assigned a negative value.

During a **dynamic simulation**, the cell densities are allowed to change until some threshold is reached for the deviation from equilibrium. Then the cell geometries are adapted so that the densities relax towards equilibrium.

2-D geometries are defined by polygons in the x - z plane. The polygons are defined by vertices and their connectivity (for details see Section 5.2). Since these specifications can be lengthy, they can be put into a separate geometry file. 2-D geometries also may be specified to be periodic in x direction. Internally, the geometry information is converted into a distance array for each region. The distance values are defined on a mesh covering the respective region.

3-D rotational symmetric geometries may be realized by combining 2-D geometries with cylindrical symmetry. For a symmetry axis parallel to the x axis, the 3-D geometry is defined by rotating the polygons about this axis. Other orientations of the symmetry axis are realized by permutations of x , y , and z [23].

Finally, a simple **surface roughness** model is provided for both 1-D and 2-D simulations. This so-called density gradient model [20] reduces the target density linearly towards the surface in a sub-surface layer. The model has been introduced to provide more realistic sputter yields at glancing incidence.

parameter	IMSIL name	to be specified in record
number of regions	NR	&SETUP
dimensionality	NDIM	&SETUP
1-D surface/interface coordinates	POSIF	&GEOM
cylindrical symmetry flag	LCYLX	&GEOM
	LCYLY	&GEOM
	LCYLZ	&GEOM
spherical symmetry flag	LSPHER	&GEOM
position of symmetry line/center	CENTER	&GEOM
number of ions for cell update	NIONUPD	&GEOM
threshold change for cell update	FRACUPD	&GEOM
fixed coordinate in cell update	POSFIX	&GEOM
2-D geometry file	FILGEOM	&GEOM
polygon vertices	POS	&GEOM
polygon connectivity	POINTS	&GEOM
periodicity flag	LPERX	&GEOM
periodic boundary positions	XPER	&GEOM
box width of distance grid	WBOX2	&GEOM
maximum size of distance grid	NBOX2	&GEOM
thickness of roughness layer	WROUGH	&GEOM

Chapter 3

Physical Models

In this chapter an overview of the physical models implemented in IMSIL is given. More details are found in the references listed at the end of this manual. A particular objective of this chapter is to give correspondence lists of the notation of the most important model parameters as used in this chapter or in the references and the names of the variables as they have to be specified in the input file. The complete information about all input parameters is found in Chapter 5.

3.1 Selection of Collision Partners

In the binary collision approximation, the collisions are treated as if the ions interacted with only one target atom at a time. In **amorphous** targets, the algorithm used is similar to that implemented in TRIM [25], however, with a different choice of the maximum impact parameter. p_{\max} is calculated depending on energy and ion and target atom species, based on two criteria [12]. The energy transfer criterion guarantees that scattering events with energy transfers larger than a user-specified value T_{\min} are taken into account. The scattering angle criterion guarantees that all scattering events with scattering angles larger than the user-specified value ψ_{\min} are taken into account. The resulting p_{\max} values are limited by $p_{\max,\max}$ and $p_{\max,\psi}$, respectively. The larger of the maximum impact parameters determined from the two criteria is taken. In addition, the impact parameter is forced to be at least $p_{\max,\min}$. It is possible to use a fixed maximum impact parameter by setting all of $p_{\max,\max}$, $p_{\max,\psi}$, and $p_{\max,\min}$ to the desired value.

The user can choose between a deterministic and a random free flight path. The latter assumes an exponential probability density function with a mean value equal to the deterministic free flight path. The (mean) free flight path \bar{L} is calculated from the relation $\bar{L} \cdot \pi p_{\max}^2 = N^{-1}$, which considers the correct atomic density N of the target. As usual [25], the square of the impact parameter p is selected randomly with a uniform distribution between 0 and p_{\max}^2 , and the azimuthal position of the target atom with respect to the ion path is chosen randomly between 0 and 2π .

In a **crystalline** target region, after each collision, the next target atom to be hit has to be selected from the lattice positions surrounding the ion. This is done among all lattice sites that lead to impact parameters $p < p_{\max,c}$ and free flight paths $0 < L < L_{\max,c}$. In addition, to limit the size of the search cell, only atoms with a distance from the projectile less than the search distance $r_{\min,c}$ are guaranteed to be checked. For consistency, the following condition must hold:

$$L_{\max,c}^2 + p_{\max,c}^2 \leq r_{\min,c}^2 \quad (3.1)$$

For the diamond crystal structure, it is usually not necessary to change the default values of these parameters. If it is done, it is important to obey Eq. (3.1). For other crystal structures, it might be essential to set $p_{\max,c}$ in the input file. With larger $p_{\max,c}$, more distant collision partners are taken into account, which might increase accuracy at low energies on the expense of increasing computation times. Too small $p_{\max,c}$ might introduce inaccuracies into the simulation. For crystal structures other than the default diamond structure, $r_{\min,c}$ is calculated by the program. The choice of $L_{\max,c}$ is uncritical, it only affects the computational efficiency.

The default values of $p_{\max,\psi}$ and $p_{\max,c}$ are implant energy dependent for energies less than 10 keV in order to allow more accurate simulation of low-energy implantations.

Small displacements from the lattice positions due to thermal vibrations are taken into account. These are assumed to be uncorrelated and distributed according to a Gaussian function. The RMS displacement in one direction, x_{rms} , may be calculated using the Debye theory [26]. A value of 490 K is recommended for the Debye temperature of Si [27, 28]. The values of the Debye temperatures for all elements are tabulated within IMSIL and appropriate vibrational amplitudes are calculated using the Debye function [26].

It has been shown that “simultaneous collisions” are not beneficial for dopant profile calculation [29, 30, 31]. “Simultaneous collisions” refer to an algorithm which treats collisions with several target atoms simultaneously, when the collisions are separated by a free flight path smaller than a (small) value L_{sim} [32]. This treatment can only be approximate, since the three-body problem cannot be solved exactly, as is known from classical mechanics. To our experience, no improvement can be obtained by using “simultaneous collisions”. This feature has therefore been eliminated from IMSIL. The quality of the treatment of collisions in crystalline targets is also supported by the good results obtained in round robin simulations [33] and other comparisons with MD simulations [17, 34].

The motion of the ions near **interfaces** between regions and between a region and vacuum is carefully treated. For this purpose, when the ion moves near to an interface, potential collision partners from both adjoining regions are considered. This is particularly important for dynamic simulations (LDYN=T) where the composition and density of the target may change from cell to cell with the cells typically being quite small. In a dynamic simulation, the implanted ions are recorded and taken into account as potential collision partners for subsequent ions and recoils.

parameter	IMSIL name	to be specified in record
dynamic simulation flag	LDYN	&SETUP
ψ_{\min}	PSIFFP	&SNPAR
T_{\min}	DENFFP	&SNPAR
$p_{\max,\min}$	PMAXMIN	&SNPAR
$p_{\max,\max}$	PMAXMAX	&SNPAR
$p_{\max,\psi}$	PMAXPSI	&SNPAR
N	DENSITY	&MATERIAL
$p_{\max,c}$	PMAX	&CRYSTAL
$L_{\max,c}$	FFPMAX	&CRYSTAL
$r_{\min,c}^2 - 1$	NRAD2	&CRYSTAL
x_{rms}	VIB	&MATERIAL

3.2 Interatomic Potential

The standard model of the interatomic potential used in IMSIL is the universal ZBL potential [35]. It reads

$$V(r) = \frac{Z_1 Z_2 e^2}{r} \cdot \Phi\left(\frac{r}{a_{12}}\right), \quad \Phi(R) = \sum_{i=1}^4 a_i \cdot \exp(-b_i R) \quad (3.2)$$

Z_1 and Z_2 denote the atomic numbers of the ion and the target atom, respectively, r the interatomic separation, and a_{12} the screening length

$$a_{12} = \frac{0.468 \text{ \AA}}{Z_1^{0.23} + Z_2^{0.23}} \quad (3.3)$$

The coefficients of the screening function Φ are given by $a_i = \{0.1818, 0.5099, 0.2802, 0.02817\}$ and $b_i = \{3.2, 0.9423, 0.4029, 0.2016\}$ [35]. The scattering angles and the energy transfer from projectile to recoil are calculated, as usual, from the scattering angle in the center-of-mass system θ . In the default model, θ is calculated by bicubic interpolation in tables of $\cot(\theta/2)$ [36]. Such tables may be set up, in principle, for any interatomic potential. However, the screening length is assumed according to Eq. 3.3 in any case. The scattering table file corresponding to the universal ZBL potential is called **SCATTAB**.

The universal ZBL screening function has been obtained by averaging a large number of “pair-specific” interatomic potentials (i.e. for specific projectile-recoil atom combinations) [35]. Tables for some of the pair-specific potentials are also provided with the program.

Alternatively, the scattering angle θ in the center-of-mass system can be determined by numerical integration using Gauss-Legendre or Gauss-Mehler quadrature. The screening function may either be assumed as a sum of exponentials (Eq. 3.2) or may be defined by a table. Tables for the screening functions of all pair-specific potentials are provided.

parameter	IMSIL name	to be specified in record
Z_1	derived from ion name	–
M_1	derived from ion name or MASS	&ATOMS
Z_2	derived from material name	–
M_2	derived from material name or MASS	&ATOMS
scattering table file	SCAFIL	&SNPAR
screening coefficients	COEFFIL	&SNPAR
screening function table	SCRFIL	&SNPAR

The format of the files SCAFIL, COEFFIL, and SCRFIL is described in Section 4.1.

3.3 Implantation Damage

The number of atoms displaced from their lattice sites may be calculated in IMSIL either using the modified Kinchin-Pease model [37] or by following the recoils. In the latter case, the recoils may be followed until they come to rest, or until their energy falls below a user-defined value E_{lim} . When the recoil energy falls below E_{lim} , the number of interstitials ν_{gen} generated in the remaining part of its collision cascade is calculated by the modified Kinchin-Pease model.

In a recoil cascade, when the energy transferred to a target atom (recoil) exceeds its displacement energy E_d , it is set into motion, and a vacancy is generated (which may be subject to recombination). When the recoil comes to rest, an interstitial is generated (which may be subject to recombination as well), unless it produces a recoil of the same type in its last collision (this is called a “replacement collision”).

Recombination with point defects generated within the same recoil cascade is treated by the recombination factor f_{rec} . Recombination with point defects generated by previous recoil cascades is taken into account with a probability proportional to the point defect concentration. When the recoil motion is followed until the end of the trajectory, the damage increments are calculated in the following way [38]:

- (i) When an interstitial is recoiled, the local interstitial number decreases by one. No other processes are assumed to be induced.

$$\Delta n_{\text{I}} = -1 \quad \Delta n_{\text{V}} = 0 \quad (3.4)$$

- (ii) When a lattice atom is recoiled, the model takes into account defect recombination both with interstitials generated within the same cascade (described by f_{rec}) and with interstitials originating from previous cascades (described by the probability $1 - kN_{\text{I}}$ that the vacancy is not located within the capture radius of an interstitial).

$$\Delta n_{\text{V}} = f_{\text{rec}}(1 - kN_{\text{I}}) \quad \Delta n_{\text{I}} = \Delta n_{\text{V}} - 1 \quad (3.5)$$

The second relation is implied by the conservation of particle numbers during recombination.

- (iii) When a recoil comes to rest, it is only allowed to recombine with vacancies from previous cascades (described by the factor $1 - kN_V$), but not with those of the same cascade. The latter would mean to count recombination processes within the recoil cascade twice and has to be excluded therefore.

$$\Delta n_I = 1 - kN_V \quad \Delta n_V = \Delta n_I - 1 \quad (3.6)$$

k is set equal to $1/N_{\text{sat}}$ if recombination with defects from previous cascades is assumed to take place, where N_{sat} denotes the saturation density of the point defects. $k = 0$ is used otherwise. Notice that n denotes defect numbers, while N denotes defect concentrations.

The above formulae assume that the additional Frenkel pairs described by f_{rec} are generated where a lattice atom is recoiled. It is also possible that part or all of the additional Frenkel pairs are generated where the recoil stops. Introducing the fraction of these additional Frenkel pairs generated at the starting point of the recoil trajectory as the parameter f_{rg} , Eqs. (3.5) and (3.6) are replaced by

$$\Delta n_V = (1 - f_{\text{rg}} + f_{\text{rg}}f_{\text{rec}})(1 - kN_I) \quad \Delta n_I = \Delta n_V - 1 \quad (3.7)$$

and

$$\Delta n_I = (f_{\text{rg}} + (1 - f_{\text{rg}})f_{\text{rec}})(1 - kN_V) \quad \Delta n_V = \Delta n_I - 1 \quad (3.8)$$

respectively. Note that these equations reduce to Eqs. (3.5) and (3.6) for $f_{\text{rg}} = 1$.

When the modified Kinchin Pease model is used to calculate the number of point defects generated in the remaining subcascade, a formula based on the above assumptions is used taking the probability of the three events into account.¹

Values of f_{rec} larger than unity indicate that there is in fact no damage recombination but rather damage multiplication due to nonlinear effects. Damage recombination and damage saturation are usually observed only for light ions. IMSIL also offers the possibility to turn the target into the amorphous state when the damage density exceeds a user specified value N_{amo} .

Values of f_{rec} , N_{sat} , N_{amo} have been determined for B [39], P [38, 40], As [38, 41], and BF_2 [38, 42] in Si. Further investigations of the damage model are reported in [28, 43]. Regarding B in Si, investigation of ultra-low energy B implants led to the conclusion that a higher value of f_{rec} and no damage saturation is preferable at low energies [30]. These values are also consistent with the dose dependence of higher-energy implantations in (100)-Si. The default values in IMSIL have been adjusted accordingly. It should be

¹The formula is similar to

$$\nu_{\text{st}} = \nu_{\text{gen}} \cdot f_{\text{rec}} \cdot (1 - N_V/N_{\text{sat}})$$

which was used in earlier versions of IMSIL. $\nu_{\text{st}} (= \Delta n_I = \Delta n_V)$ denotes the average number of stable point defects.

noted, however, that one should revert to the values given in [44] if the dose dependence of [110] channeling is investigated.

As an alternative in case of true recombination ($f_{\text{rec}} < 1$), the capture radius model may be used [43]. In this model the positions of vacancies and interstitials generated in each collision cascade are stored, and vacancies and interstitials that are within a capture radius r_{cap} are annihilated.

As an alternative in case of damage multiplication ($f_{\text{rec}} > 1$), the amorphous pocket model may be used [45]: In this model, each energy deposition event is assigned a sphere the volume of which would be molten according to the heat of fusion (0.52 eV/atom in Si), if surface effects could be excluded. After each collision cascade, the connected volume of overlapping melting spheres are determined. Each such overlapping area defines an amorphous pocket candidate. The number of displaced atoms associated with it is the ratio of the sum of energies deposited by their members and the heat of fusion. Surface effects are taken into account by shrinking the amorphous pocket candidate by a certain amount. If the remaining pocket meets a minimum size requirement, an amorphous pocket is generated with the number of displaced atoms calculated as indicated above. Otherwise the amorphous pocket candidate is abandoned, and the interstitials contained in it are used to calculate the number of displaced atoms.

In addition to the above model, an ion beam induced interfacial amorphization (IBIIA) model [46] can be activated. In this model, existing amorphous zones grow with a rate proportional to nuclear energy deposition.

parameter	IMSIL name	to be specified in record
displacement energy E_d	ED	&DAMAGE
recoil cutoff energy E_{lim}	ELIMKP	&DAMAGE
replacement collision flag	LREPL	&DAMAGE
recombination fraction f_{rg}	FRACRG	&DAMAGE
damage recombination factor f_{rec}	FREC	&DAMAGE
damage saturation density N_{sat}	DAMSAT	&DAMAGE
amorphization threshold N_{amo}	DAMAMO	&DAMAGE
capture radius model flag	LCAP	&DAMAGE
capture radius	RCAP	&DAMAGE
amorphous pocket model flag	LPOCK	&DAMAGE
heat of fusion	EMELT	&DAMAGE
amorphous pocket shrink parameter	FPOCK	&DAMAGE
minimum pocket size	NPOCKMIN	&DAMAGE
IBIIA model flag	LIBIIA	&DAMAGE
IBIIA growth rate constant	RIBIIA	&DAMAGE

3.4 Backscattering, Transmission, and Sputtering

Ions leaving the target on the same side as they entered it, are called **backscattered** or reflected. Ions leaving a membrane on the opposite side of where they entered it, are called transmitted. In a conventional 1-D membrane setup, the ions impinge on the top surface (lowest z coordinate of the target) and may be emitted from that surface with a negative z component of their direction of motion. Conversely, if they are emitted from the bottom surface, they have a positive z component of their direction of motion. IMSIL takes the sign of the z component of the direction of motion as the criterion for distinguishing between “backscattered” and “transmitted” atoms. Note that this distinction may not always make sense, e.g., when ions can be emitted from the sides of a 2-D or 3-D target. In those cases it might be more reasonable to just add backscattered and transmitted ions.

Target atoms that are followed as recoils and leave the target, are called **sputtered**. They are also classified into backscattered and transmitted recoils, which corresponds to backward and forward sputtered atoms. There is no difference in treating ions and recoils with respect to their leaving the target. Therefore, their statistics is always reported alongside.

When accurate sputtering characteristics are desired, the “accurate sputtering flag” should be set. Then recoils are followed down to the surface binding energy E_{surf} rather than down to the displacement energy E_d , if they are closer to the surface than some distance d_{near} . Within this distance, recoils with energies less than the displacement energy are followed as “virtual” recoils. Virtual recoils deposit energy like real recoils and are counted as sputtered when they leave the target, but they generate defects (vacancies at the starting point, and interstitials at the end point if the recoil reenters the target) only when they leave the target.

With the accurate sputtering flag set, also the maximum impact parameter p_{max} is adjusted in the near-surface zone such that no collisions with energy transfer larger than the surface binding energy are missed.

When the ions enter the target, they gain the surface binding energy, and ions and recoils loose the surface binding energy when they are emitted. Depending on the model of the potential barrier, they may also be refracted. A planar potential barrier [47] is usually a good choice and is therefore the default. The potential barrier model can be changed to isotropic, where there is no refraction, or to an intermediate model [19] using a “nonplanarity parameter”.

When a recoil reaches the surface with an energy not sufficient to surmount the potential barrier, it is reflected. With the planar potential model, it may happen that the recoil bounces back and forth between the barrier (which is at p_{max} outside the target) and the target. The number of reflections may be limited.

parameter	IMSIL name	to be specified in record
accurate sputtering flag	LSPUT	&SETUP
surface binding energy E_{surf}	ESURF	&DAMAGE
displacement energy E_d	ESURF	&DAMAGE
d_{near}	DNEAR	&DAMAGE
nonplanarity parameter	KSURF	&DAMAGE
maximum number of reflections	REFLECTMAX	&DAMAGE

3.5 Electronic Stopping

The electronic stopping of channeled ions is reduced as compared with ions moving in a random direction. This may be taken into account by an impact parameter dependent model. The model implemented in IMSIL is composed of a part which is proportional to the path length L

$$\Delta E_e^{\text{nl}} = N \cdot S_e \cdot L \cdot \left[x^{\text{nl}} + x^{\text{loc}} \cdot \left(1 + \frac{p_{\text{max}}}{a} \right) \cdot \exp \left\{ -\frac{p_{\text{max}}}{a} \right\} \right] \quad (3.9)$$

and a part which depends on the impact parameter p

$$\Delta E_e^{\text{loc}} = x^{\text{loc}} \cdot \frac{S_e}{2\pi a^2} \cdot \exp \left\{ -\frac{p}{a} \right\} \quad (3.10)$$

N denotes the atomic density of the target. The impact parameter dependence is according to Oen and Robinson [48]. Alternatively, the impact parameter dependence can be chosen by a generalized Firsov model [49], in which case the model reads

$$\Delta E_e^{\text{nl}} = N \cdot S_e \cdot L \cdot \left[x^{\text{nl}} + x^{\text{loc}} \cdot \frac{1 + (d-1) \frac{p_{\text{max}}}{a}}{\left(1 + \frac{p_{\text{max}}}{a} \right)^{d-1}} \right] \quad (3.11)$$

and

$$\Delta E_e^{\text{loc}} = x^{\text{loc}} \cdot \frac{S_e}{2\pi a^2} \cdot (d-1)(d-2) \left(1 + \frac{p}{a} \right)^{-d} \quad (3.12)$$

In the original Firsov model $d = 5$. We do not use Firsov's prefactor and screening length. If desired, Firsov's values can be set by appropriate values of k/k_L and f (see below).

In Eqs. (3.10) and (3.12), the distance of closest approach r_0 in the collision (apsis of the collision) may be used instead of the impact parameter p . The most significant effect of this choice is a reduction in electronic stopping at low energies.

a is the screening length of the impact parameter dependent part, which is expressed by the value $a_{\text{OR}} = a_{\text{ZBL}}/0.3$ proposed by Oen and Robinson [48] with a_{ZBL} the screening length of the ZBL interatomic potential [35].

$$a = f \cdot a_{\text{OR}} (= f \cdot a_{12}/0.3) \quad (3.13)$$

The random electronic stopping power S_e may be calculated in either of two ways. In the first case (models # 1 or #3) it is assumed to be a power of the ion energy

$$S_e = S_{eL} = k \cdot E^p \quad (3.14)$$

With $p = 0.5$ this is Lindhard's model [50], where electronic stopping is proportional to the ion velocity. In order to describe the electronic stopping power for high-energy ions, in model # 3 a formula similar to that of Biersack [25] is used [9]

$$S_e = \left(S_{eL}^{-c} + S_{eB}^{-c} \right)^{-1/c} \quad (3.15)$$

where S_{eL} denotes the Lindhard like stopping power according to Eq. (3.14) and S_{eB} the slightly modified Bethe Bloch stopping power as described in [25]:

$$S_{eB} = \frac{B}{E_b} \cdot \ln \left(E_b + c_0 + \frac{c_1}{E_b} \right) \quad (3.16)$$

where B and E_b denote ion-target dependent prefactors. (For details see [25]). In the second case (models # 2 or # 4) the ZBL electronic stopping power [35] is used. Alternatively, the random electronic stopping power may also be read from a file.

x^{nl} and x^{loc} denote the nonlocal (L dependent) and the local (p dependent) fraction of S_e , respectively.

$$x^{\text{nl}} + x^{\text{loc}} = 1 \quad (3.17)$$

It has been found that the nonlocal fraction x^{nl} is energy dependent. The energy dependence can be specified by specifying x^{nl} at different energies E^{nl} . The actual values are interpolated according to

$$x^{\text{nl}} \sim E^q \quad (3.18)$$

(models #1 and #2), or

$$x^{\text{nl}} \sim S_e(E)^{2q} \quad (3.19)$$

(models #3 and #4) where q is determined by the program from x^{nl} values at two energies. Note that Eq. (3.19) reduces to Eq. (3.18) when the Lindhard like stopping power Eq. (3.14) is used. Default values of the parameters are provided by the program for all ions in silicon [1, 10, 39, 41, 51, 52]. The value of k is specified by a correction factor k/k_L to the Lindhard stopping power.

Electronic energy loss straggling may be considered according to Konac, Klatt, and Kalbitzer [52, 53].

parameter	IMSIL name	to be specified in record
target density N	DENSITY	&MATERIAL
model number	MODEL	&SEPAR
Lindhard correction factor k/k_L	CORLIN	&SEPAR
power p in Lindhard stopping	POWLIN	&SEPAR
interpolation power c	POWINT	&SEPAR
Bethe stopping constant c_0	COBETHE	&SEPAR
Bethe stopping constant c_1	C1BETHE	&SEPAR
random electronic stopping file	FILSE	&SEPAR
nonlocal fraction x^{nl}	XNL	&SEPAR
energy for x^{nl} , E^{nl}	ENL	&SEPAR
screening length factor f	FACSCR	&SEPAR
Firsov p dependence flag	FIRSOV	&SEPAR
Firsov p dependence power d	POWFIRS	&SEPAR
apsis flag	LAPSIS	&SEPAR
electronic straggling flag	STRAGGLE	&SEPAR
electronic straggling kink energy	ESTRAG	&SEPAR

3.6 Rutherford Backscattering / Channeling

The code uses the principle of close encounter probability [54] and approximates the paths of the backscattered particles by straight trajectories in a random medium. The program gives as the output the RBS signal as a function of depth or backscattering energy. The nuclear encounter probability is the cross section for the process times the integral of the product of the ion flux and the probability of a target atom being at the point of passage.

The Rutherford backscattering cross section relates collision densities to target atom densities. The probability of the collision is proportional to the cross section area of the target sphere perpendicular to the path of the projectile. The resulting probability that an ion will be scattered into a detector with a given solid angle is represented in derivative form, the derivative of the cross-section with respect to a differential solid angle evaluated at a particular solid angle and projectile energy. The resulting Rutherford differential cross section is given by:

$$\frac{\partial\sigma}{\partial\Omega} = \left[\frac{Z_1 Z_2 e^2}{2E} \right]^2 \cdot \frac{1}{\sin^4 \psi} \cdot \frac{\left[\sqrt{1 - \left(\frac{m_1 \sin \psi}{m_2} \right)^2} + \cos \psi \right]^2}{\sqrt{1 - \left(\frac{m_1 \sin \psi}{m_2} \right)^2}} \quad (3.20)$$

$e^2 = 1.44 \times 10^{-13}$ MeV·cm is constant, E and m_1 are the energy and the mass of the incident ion, respectively, m_2 is the mass of the target atom, and ψ is the laboratory backscattering angle. As the result of a backscattering event, a particle will loose energy. The ratio of the projectile energy immediately after scattering to the energy immediately

before scattering is the kinematic factor:

$$K = \left[\frac{\sqrt{1 - \left(\frac{m_1 \cdot \sin \psi}{m_2}\right)^2} + \frac{m_1 \cdot \cos \psi}{m_2}}{1 + \frac{m_1}{m_2}} \right]^2 \quad (3.21)$$

Atoms are assumed to vibrate independently, where the displacement probability is Gaussian distributed,

$$P = \frac{1}{2\pi x_{\text{rms}}^2} \cdot \exp\left(-\frac{r^2}{2x_{\text{rms}}^2}\right), \quad (3.22)$$

where x_{rms} denotes the RMS vibration amplitude of the target atoms in one direction, and r the distance vector between collision point and target atom. The energy detected in the detector (the backscattered energy) is calculated by approximating the path of the backscattered particle by a straight line in random medium. The backscattering energy is scored in an energy spectrum:

$$P_{\text{RBS}} = \frac{\partial \sigma}{\partial \Omega} \cdot P \cdot N_{\text{d}} \quad (3.23)$$

where N_{d} is the ion dose. The units of P_{RBS} are $\text{cm}^{-2}\text{sr}^{-1}$.

parameter	IMSIL name	to be specified in record
backscattering angle ψ	TILTA	&OUTPUT
dose N_{d}	DOSE	&IONS
thermal vibration amplitude x_{rms}	VIB	&MATERIAL

Chapter 4

Files

In this chapter, all data files which are required or produced by IMSIL are listed, and their format is described. If the Python scripts delivered with IMSIL are used, the file `xxx.inp` is copied to the directory `xxx` and renamed to `INP`. The same is done with all other files `xxx.*`. After completion of an IMSIL run, all files `EXT` in directory `xxx` (where `EXT` stands for an arbitrary file name) are copied to the parent directory and renamed to `xxx.ext`. E.g., if `example1.inp` is an input file and the `imsil` script is run with `python <path-to-imsil.py> example1.inp`, then a subdirectory `example1` is created and the file `example1.inp` is copied to `example1/INP`, the current directory is changed to `example1`, and IMSIL is run. The simulation generates an output file `OUT`, which is copied to the parent directory as `example1.out`. The same is done with any other files that are created during the run.

When the Python script `imsil.py` is used, IMSIL is run in a subdirectory. Thus, when referring in the input file to a file in the current directory, you have to prepend `..` to the file name, e.g., `FILHIS='../example2.his'` on a Linux machine. If execution is delegated to a remote computer, these files have to be copied by hand. An exception are the files that are provided with IMSIL in the `tables` directory. Their path must be specified relative to the `tables` directory or one of its subdirectories (see the file descriptions).

4.1 Input Files

There are the following input files:

(Default) Name	File contents, remarks	file name specified by parameter	in record
INP	Input parameters	—	—
(SCATTAB)	Scattering table file (used if QUAD='no' in record &SNPAR)	SCAFIL	&SNPAR
(ZBL)	Screening function file (used if QUAD≠'no' and if SCRFUN='scrfil' in record &SNPAR)	SCRFIL	&SNPAR
(ZBLspec)	Screening coefficient file (used if QUAD≠'no' and if SCRFUN='coeffil' in record &SNPAR)	COEFFIL	&SNPAR
(-)	Random electronic stopping file (used if MODEL=7 in record &SEPAR)	FILSE	&SEPAR
(-)	Definition of the crystal lattice unit cell	NAME	&CRYSTAL
(BCK)	Backup = data necessary to continue an old run (used if USEBK=T in record &SETUP)	BKFIL	&SETUP
(-)	Histogram = histogram profiles necessary to initialize 1-D histogram of desired atom index (used if USEHIS=T in record &SETUP)	FILHIS	&SETUP
(-)	Cell file = 1-D cell-based description of the target containing the densities of each species (used if USECELL=T in record &SETUP)	FILCELL	&SETUP
(-)	Trajectory file used to initialize the ions (used if USETRA=T)	FILTRA	&ION

The format of the INP file is described in detail in Chapter 5.

The scattering table file SCAFIL is assumed to be located in the subdirectory `scattab` of the `tables` directory and is read by the following FORTRAN statements:

```

READ (22,*) NE, NP
READ (22,*) (E(IE), IE=1,NE)
READ (22,*) (P(IP), IP=1,NP)
READ (22,*) PCUTOFF
READ (22,*) ((COTTHH(IE,IP), IE=1,NE), IP=1,NP)
READ (22,*) ((DCOTDE(IE,IP), IE=1,NE), IP=1,NP)
READ (22,*) ((DCOTDP(IE,IP), IE=1,NE), IP=1,NP)
READ (22,*) ((TAU(IE,IP), IE=1,NE), IP=1,NP)
READ (22,*) ((DTAUDE(IE,IP), IE=1,NE), IP=1,NP)
READ (22,*) ((DTAUDP(IE,IP), IE=1,NE), IP=1,NP)

```

NE and NP denote the number of tabulated reduced energy and reduced impact parameter values E and P, respectively. The reduced energy is the energy divided by

$(1 + M_1/M_2)Z_1Z_2e^2/a_{12}$, the reduced impact parameter is the impact parameter divided by a_{12} , with a_{12} the ZBL screening length according to Eq. (3.3). PCUTOFF is the screened cut-off impact parameter, beyond which the deflection angle will be set to zero. COTHH contains the tabulated values of $\cot(\theta/2)$ with θ the scattering angle in the center-of-mass system. DCOTDE and DCOTDP are the derivatives of $\cot(\theta/2)$ with respect to reduced energy and reduced impact parameter, respectively. TAU denotes the scaled time integral, DTAUDE and DTAUDP its derivatives.

The screening function file SCRFIL is assumed to be located in the subdirectory `screentab` of the `tables` directory and is read by

```
DO I=1,N
  READ (22,*) R(I), PHI(I)
END DO
```

N denotes the number of data points, R the internuclear distance in Ångströms, and PHI the value of the screening function $\Phi(r)$, see Eq. (3.2). If SCRFIL='ZBL', then the element specific ZBL screening functions provided in the subdirectory `screentab/zbl` of the `tables` directory are used.

The screening coefficient file COEFFIL is assumed to be located in the subdirectory `screencoefs` of the `tables` directory and is read by

```
DO
  READ (22,*) Z1, Z2, (A(I,J), B(I,J), I=1,4)
END DO
```

until the desired atomic numbers Z1 and Z2 are found. A and B are the coefficients a_i and b_i , respectively, of Eq. (3.2). Note that the number of terms in Eq. (3.2), when used with COEFFIL, must be 4.

The random electronic stopping power file FILSE provides an option to include measured stopping power data, or data obtained by an external program, in the simulation. It is read by the following FORTRAN statements:

```
DO, IE=0,NE
  READ (122,*) E(IE), DEDX(IE)
ENDDO
```

The file can begin with an arbitrary number of comment lines beginning with the `*` character, which will be ignored when the file is read. E denotes the energy [eV]. DEDX is the average energy loss per path length [eV/Å]. NE denotes the number of (E,DEDX) points (maximum 300).

The file defining a crystal structure (NAME parameter of the `&CRYSTAL` record) is read by the following FORTRAN statements:

```
READ (123,*) ALAT(1:3)
READ (123,*) NSITE
DO, ISITE=1,NSITE
```

```
      READ(123,*) X(ISITE), Y(ISITE), Z(ISITE), IELEM(ISITE)
    ENDDO
```

`ALAT(1:3)` denote the lattice constants (edge lengths of the unit cell), `NSITE` the number of lattice sites, and `X`, `Y`, `Z` the positions of the lattice sites. $0 \leq X \leq \text{ALAT}(1)$, $0 \leq Y \leq \text{ALAT}(2)$, and $0 \leq Z \leq \text{ALAT}(3)$ must hold. `IELEM` denotes the element number and must equal 1 for a pure element or integers ranging from 1 to the number of elements in a compound. The lattice constants and the lattice site coordinates will be scaled such that the correct material density as specified by the `DENSITY` parameter of the `MATERIAL` record is obtained.

The backup file `BKFIL` stems from the old days, when simulations might have taken weeks. Its purpose was to allow the continuation of interrupted simulation. It may still be useful for debugging purposes. New simulations should not rely on the `BKFIL`, since its full functionality is not guaranteed to be maintained in the future. Note that histograms may be read in from a histogram file with the `FILHIS` parameter of the `&SETUP` record. The format of the backup file is binary.

The other possible input files are also output files of `IMSIL`. Their format is described in Sections 4.3 and 4.6.

4.2 The Output File OUT

In the first part of the OUT file, most of the input parameters, including those which have not been specified in the INP file and therefore assume their default values, and some internal parameters are listed. In the second part, after the header “SIMULATION RESULTS”, statistical output data are listed. This starts with energy deposition and energy transfer statistic. Energy **transfer** means energy transfers from a projectile to a recoil that is set into motion. **Deposition** denotes all other kinds of energy transfer or loss by a projectile. This includes electronic energy loss (“Electrons”, called “ionization” in TRIM [55]), nuclear energy deposition (“Phonons+Potential”, called “phonons” in TRIM), and the total energy of backscattered and transmitted projectiles.

For the purpose of this manual, we define **yield** as the average amount of the quantity of interest for one impinging ion. “Yield” is commonly used in “sputtering yield”, which means the number of sputtered atoms per incident ion. Similarly, we define the yield of stopped atoms (ions or recoils) and vacancies, of energy deposition, and of backscattered and transmitted projectiles. The yield of deposited energy, e.g., is the total deposited energy per ion.

Normally, one is interested only in the mean value $\langle Y \rangle$ of the yield. However, the yields Y_i are different for each impinging ion i , so there is a yield distribution, which can be described by **moments**. The moments are defined by

$$\text{mean value : } \quad \langle Y \rangle = \frac{1}{N} \sum_{i=1}^N Y_i \quad (4.1)$$

$$\text{standard deviation : } \quad \sigma_Y = \left[\frac{1}{N} \sum_{i=1}^N (Y_i - \langle Y \rangle)^2 \right]^{1/2} \quad (4.2)$$

$$\text{skewness : } \quad \gamma_Y = \frac{1}{N} \sum_{i=1}^N (Y_i - \langle Y \rangle)^3 / \sigma_Y^3 \quad (4.3)$$

$$\text{kurtosis : } \quad \beta_Y = \frac{1}{N} \sum_{i=1}^N (Y_i - \langle Y \rangle)^4 / \sigma_Y^4 \quad (4.4)$$

These moments are listed in the first four lines following a header. In the following four lines, the $1\text{-}\sigma$ standard deviations (corresponding to the statistical uncertainties) of the moments are listed.

The energy statistics and the yield moments are always written to the OUT file. Other moments are written if requested in the input file, see the LMOM** parameters of the &OUTPUT record. In the moment definitions for other quantities than the sputter yield, Y has to be replaced in Eqs. (4.1)–(4.4) by the quantity of interest. E.g., for the moments of a “vertical” distribution, Y_i has to be replaced by the coordinate z_i of the stopped projectile. Note that what is commonly called the projected range of the implanted ions, is listed under Spatial moments of stopped atoms and vacancies/Vertical moments/mean value, the longitudinal straggling under Spatial moments of stopped atoms and vacancies/Vertical moments/std.dev., and the lateral straggling under Spatial moments of stopped atoms and vacancies/Lateral x-moments/std.dev. and/or Spatial moments of stopped atoms and vacancies/Lateral y-moments/std.dev.. These and other parameters can be extracted using the read_out function of the read_output.py module provided with IMSIL (see Section 6).

4.3 1-D Histogram Files

For the purpose of this manual we define the term **distribution** as the probability density function times the yield times the implantation dose. As an example, the nuclear energy deposition distribution $f_{\text{NE}}(z)$ is the probability density function of nuclear energy deposition times the total nuclear energy deposited in the target per ion, Y_{NE} , times the area dose DOSE as specified on the &IONS record. Since probability density functions are normalized,

$$\int_{-\infty}^{\infty} f_{\text{NE}}(z) dz = Y_{\text{NE}} \times \text{DOSE} \quad (4.5)$$

Internally, IMSIL's length units are Ångströms. Thus, an area dose DOSE as specified in the INP file is converted to Å^{-2} , and the units of $f_{\text{NE}}(z)$ are $\text{eV}/\text{Å}^3$. Note that the units of Y_{NE} are eV. eV and Ångströms, as well as degrees for angular distributions, are generally used for the units of the histograms. There is one exception to this rule, spatial atom distributions in the HIS file (also known as atom concentrations) are converted to cm^{-3} when written to the file.

Eq. (4.5) can also be written

$$\int_{-\infty}^{\infty} \frac{f_{\text{NE}}(z)}{\text{DOSE}} dz = Y_{\text{NE}} \quad (4.6)$$

When the default implantation dose of $\text{DOSE} = 10^{16} \text{ cm}^{-2} = 1 \text{ Å}^{-2}$ is used, $f_{\text{NE}}(z) = f_{\text{NE}}(z)/\text{DOSE}$ numerically. Therefore, with the default dose the distributions, except for the atom concentrations on the HIS file, may also be considered the probability density functions times the yield. E.g., in the case of nuclear energy deposition, $f_{\text{NE}}(z)$ may be interpreted as the nuclear energy deposition per ion and width Δz of the depth interval (units: $\text{eV}/\text{Å}$).

The above description is valid if DOSE refers to an area dose. With DOSEUNITS one may also demand DOSE to be interpreted as a line dose (DOSEUNITS='cm-1') or an ion number count (DOSEUNITS='1'). In these cases, the area dose is calculated using the extent of the implant area as defined by XINIT and YINIT on the &IONS record. If the extent of the implant area in one of the directions is zero, it is replaced by unity. As a consequence, the units of $f_{\text{NE}}(z)$ will be different. For instance, when a line dose is specified and XINIT(1)=XINIT(2), the area dose cannot be calculated and the line dose substituted in Eq. (4.5) for DOSE will have units Å^{-1} . As a result, the units of $f_{\text{NE}}(z)$ will be $\text{eV}/\text{Å}^2$. If the energy density per Å^3 is to be calculated a-posteriori, it is obtained by dividing $f_{\text{NE}}(z)$ by the extent of the implant area in x direction. However, these values will represent only the deposited energy or concentration inside the implant area sufficiently far from the edges. **It is therefore generally not recommended to record 1-D histograms when the dose is not specified as an area dose.** For completeness we mention that in case of atoms densities, the units will be $\text{Å}/\text{cm}^3$ in the above example.

IMSIL produces the following 1-D histogram files if the respective flag listed in the last column of the following table is set on the &OUTPUT record:

Name	File content	flag
HIS	Spatial distribution of stopped atoms	LHIS
CELL	Cell-based target description file	LCELL
HISEE	Spatial distribution of deposited electronic energy	LHISEE
HISNE	Spatial distribution of deposited nuclear energy	LHISNE
HISM	Spatial distribution of displacement moments	LHISM
HISP	Spatial distribution of deposited momentum	LHISP
HISD	Displacement distribution	LHISD
HISIV	I-V pair distance distribution	LHISIV
HISB	Spatial distribution of backscattered atoms	LHISB
HISEB	Energy distribution of backscattered atoms	LHISEB
HISAB	Angular distribution of backscattered atoms (2-D angle)	LHISAB
HISAAB	Angular distribution of backscattered atoms (3-D angle)	LHISAAB
HIST	Spatial distribution of transmitted atoms	LHIST
HISET	Energy distribution of transmitted atoms	LHISET
HISAT	Angular distribution of transmitted atoms (2-D angle)	LHISAT
HISAAT	Angular distribution of transmitted atoms (3-D angle)	HISAAT
RBS	RBS spectra	LRBS

The histogram files are generally written at the end of the simulation. They are written by the following FORTRAN statements:

```

WRITE (22,*) 1+NSPEC,2*NBOX+2
WRITE (22,*) NINT(E0), (NINT(Z1(I)), I=1, NSPEC)
WRITE (22,*) Z(1), (1E-4*HISMIN, I=1, NSPEC)
DO, IBOX=1, NBOX
  WRITE (22,*) Z(IBOX), (HIST(I, IBOX), I=1, NSPEC)
  WRITE (22,*) Z(IBOX+1), (HIST(I, IBOX), I=1, NSPEC)
ENDDO
WRITE (22,*) Z(NBOX+1), (1E-4*HISMIN, I=1, NSPEC)

```

where the meaning of the variables is detailed below.

NSPEC denotes the number of atom species to be recorded, which depends on the type of the histogram and LDYN. Let N1 denote the number of atom species in the ion (N1=1 for atomic ions) and N2 the number of recoil species to be considered in the statistics. For LDYN=T, all atom species may become recoils and must be kept track off, so N2=NATOM, where NATOM is the number of atom species specified on the &SETUP record. For LDYN=F, only the atoms of the materials may become recoils, thus N2=NATOM-N1. Note that for LDYN=T, ion atoms are contained in both N1 and N2, so one can distinguish the properties of implanted ions from those previously implanted.

NSPEC now depends on N1 and N2: In addition, for the HIS file, NSPEC depends on two parameters of the &DAMAGE record: If LRcoil=T, NSPEC=N1+2*N2. Otherwise, if LDAM=T, NSPEC=N1+N2. Else NSPEC=N1. For the HISEE, HISNE, and HISD file, NSPEC=N1+N2. For the HISM and HISP file, NSPEC=3*(N1+N2). For the HISIV file NSPEC=N2, and for the CELL

file NSPEC=NATOM.

From line 2 on, the 1-D histogram files contain NSPEC+1 columns. In the first column, the independent variable Z (depth, energy, angle) is listed, in the other columns the histogram values for the NSPEC atom species. They correspond from left to right to the ion atoms, atoms that may become recoils (and thus interstitials), and vacancies left behind by recoils. The atom species are ordered according to their occurrence in the ion and material names: First the ion, and then the materials from region 1 to how many regions are defined. Within the ion and material names from left to right. E.g., if the ion and 3 materials are defined by

```
&IONS NAME='B' ... /
&MATERIAL REGION=2 NAME='SI3N4' ... /
&MATERIAL REGION=1 NAME='SIO2' ... /
&MATERIAL REGION=3 NAME='SI' ... /
```

B is defined first, Si second, O third, and N fourth. Therefore, column 2 of the HIS file corresponds B. If LDAM=T or LRCOIL=T, column 3 corresponds to Si, column 4 to O, and column 5 to N. If LRCOIL=T, column 6 corresponds to Si vacancies, column 7 to O vacancies, and column 8 to N vacancies. To help identify the atom species, their atomic numbers are given in line 2 of the file. Atomic numbers of vacancies are given a negative sign.¹

NBOX denotes the number of histogram boxes (bins) and correspond to NBOX, NBOXA, NBOXD, or NBOXE of the &OUTPUT record (NBOX* for short). The user may choose between two options regarding how histograms evolve during the simulation. In both cases, the histograms are initialized with a box width of WBOX* of the &OUTPUT record, and are then adapted as to include all data points. When NBOX* is specified as a positive number on the &OUTPUT record (or its default value is being used), the histogram is adjusted by doubling the box width as required, while leaving the number of boxes the same. Only nonzero histogram values are written to the histogram file at the edges, so the number of boxes written to the file may be smaller than NBOX specified on the &OUTPUT record. On the other hand, if NBOX*=0 is specified on the &OUTPUT record, the box widths are kept constant equal to WBOX*, while the number of boxes is increased as required.

From line 3 on, there are 2*NBOX+2 rows. Z(IBOX) and Z(IBOX+1) are the lower and the upper boundary of histogram box IBOX, respectively, and HIST(I, IBOX) denotes the distribution function f . The physical meaning of Z and HIST depends on the file. As apparent from the code, the file contains duplicate information. This is to facilitate plotting by allowing for simply connecting the data points to draw the histogram. The first and last data point are added to provide a visual closure of the first and last box.

HIS, CELL, HISEE, HISNE, HISM, and HISP contain spatial histograms with Z denoting the depth coordinate.

¹The columns 2 to N1+N2+1 in the histogram files correspond to the atom indices as defined in Section 5.3.

In the `HIS` and `CELL` file, `HIST(I, IBOX)` denotes the concentration [cm^{-3}] of atom species `I` in histogram box `IBOX`. As described above, the `HIS` and `CELL` files differ in the atom species recorded. E.g., the `CELL` file never contains vacancy concentrations. In addition, `HIS` and `CELL` files differ in that the atom concentrations in the `CELL` file include all target atoms, while in the `HIS` file they include only atoms added in the implant step plus, optionally, those read in from a `HIS` file at the beginning of the simulation (see the `USEHIS` and related parameters of the `&SETUP` record).

In the `HISEE` and `HISNE` files, `HIST(I, IBOX)` contains the distributions of electronic and nuclear energy deposition, respectively. The units are [$\text{eV}/\text{\AA}^3$].

The `HISM` file contains the histogram of the first redistributive moment as a function of depth. The first redistributive moment is the sum of the differences between end-point and initial coordinates over all recoil trajectories that start and end inside the target [56]. Since it is a vector quantity, three columns are stored in the `HISM` file for each atom type in adjacent columns (`x`, `y`, and `z` component of the moment). The units of `HIST` are \AA .

The `HISP` file contains the histogram of the deposited momentum as a function of depth. Deposited momentum is the momentum of the ions and recoils when their simulated trajectories are terminated. Since it is a vector quantity, three columns are stored in the `HISP` file for each atom type in adjacent columns (`x`, `y`, and `z` component of the momentum). The units of `HIST` are $(m_0 \text{ eV})^{1/2}/\text{\AA}^3$, where m_0 denotes the atomic mass unit. To obtain the momentum in SI units (kg m/s), one has to multiply with $(m_0 \text{ eV/J})^{1/2} = 1.6311 \times 10^{-23}$.

The `HISD` file contains the displacement distribution, where `Z` denotes the distance between origin and end point of recoil trajectories. Similarly, the `HISIV` file contains the I-V pair distance distribution of a cascade, where `Z` denotes the distance between interstitial and vacancy [57]. The histogram is obtained by looping over both the interstitials and vacancies of each cascade and collecting the distances between them in a histogram. The I-V pair distribution function is a probability density function. The positions of the vacancies may be restricted to certain depth intervals (`POSIVMIN` and `POSIVMAX` of the `OUTPUT` record). `NHISIV` such intervals may be specified. `NHISIV` data blocks corresponding to the code snippet given above are written to the `HISIV` file.

The `HIS*B` and `HIS*T` files contain distributions of backscattered and transmitted atoms, respectively. `HISB` and `HIST` contain the spatial distributions of the ejected atoms as a function of the lateral coordinate x (i.e., in the code snippet given above, `Z` has to be replaced by `X`). The number of species listed, `NSPEC`, is the same as for the spatial atom distributions inside the target, stored in the `HIS` file. For the first `N1+N2` columns, the coordinate x refers to the position where the atom leaves the surface boundary layer, which normally is a few \AA ngstr\u00f6ms above the surface. For the last `N2` columns, x refers to the position where the ejected atoms originate from.

The `HISAB`, `HISAAB`, `HISAT`, and `HISAAT` files contain angular distributions of ejected atoms. In the `HISAB` and `HISAT` file, the angle is between the projection of the atom's direction of motion to the x - z plane and the negative and positive z axis, respectively. In the `HISAAB` and `HISAAT` file, the angle is taken between the direction of motion and the

analyzer direction defined by TILTA and ROTATA of the &OUTPUT record. The units of the angles are degrees. Depending on the LRCOIL parameter of the &DAMAGE record, data for NSPEC=N1 or NSPEC=N1+N2 atom species are written to the files.

The HISEB and HISET files contain the energy distributions of the ejected atoms. The units of the energies are eV. Depending on the LRCOIL parameter of the &DAMAGE record, data for NSPEC=N1 or NSPEC=N1+N2 atom species are written to the files.

Finally, the RBS file contains the energy and depth spectra of a Rutherford backscattering simulation (LRBS=T on the &OUTPUT record). If the target contains only one atom species (N2=1), then two blocks according to the code snippet at the beginning of this section are written, first the energy and then the depth distribution. While in these two blocks the contributions from all target atoms are added, thus each one containing two columns of data (energy/depth and the distribution, NSPEC=1), for N2>1 two other blocks are written where the contributions of different target atom species are separated (NSPEC=N2). The units of energy and depth are eV and Å, respectively. The units of the distribution functions are $\text{eV}^{-1}\text{cm}^{-2}\text{sr}^{-1}$ and $\text{Å}^{-1}\text{cm}^{-2}\text{sr}^{-1}$, respectively.

4.4 2-D Histogram Files

Name	File content	flag
HIS2	2-D spatial distribution of stopped atoms	LHIS2
HISEE2	2-D spatial distribution of deposited electronic energy	LHISEE2
HISNE2	2-D spatial distribution of deposited nuclear energy	LHISNE2
HIS2B	2-D spatial distribution of backscattered atoms	LHIS2B
HISA2B	2-D direction cosine distribution of backscattered atoms	LHISA2B
HIS2T	2-D spatial distribution of transmitted atoms	LHIS2T
HISA2T	2-D direction cosine distribution of transmitted atoms	LHISA2T

The 2-D histogram files are written by the following FORTRAN statements:

```

CHARACTER*80 TEXT
WRITE (22) TEXT
WRITE (22) NBOXZ,NBOXX,NSPEC
WRITE (22) (Z( IBOXZ) , IBOXZ=1 ,NBOXZ)
WRITE (22) (X( IBOXX) , IBOXX=1 ,NBOXX)
DO I=1,NSPEC
  WRITE (22) ((HIST(I, IBOXX, IBOXZ) , IBOXX=1 ,NBOXX) , IBOXZ=1 ,NBOXZ)
ENDDO

```

TEXT is a character string containing information about the stored histogram. NSPEC has the same meaning as for the 1-D histograms. NBOXZ and NBOXX denote the number of histogram boxes in z and x direction, respectively. Regarding adaptation of the histogram during the simulation, the same applies as for the 1-D histograms.

Z(IBOXZ) and X(IBOXX) denote the z and the x coordinates, respectively, of the center of the histogram box with indices IBOXZ and IBOXX. If LCYL=F is set on the &OUTPUT record, HIST(I, IBOXX, IBOXZ) is the concentration (energy density in case of LHISEE2=T or LHISNE2=T) of histogram I in the box (IBOXZ, IBOXX). If output in cylinder coordinates is desired (LCYL=T), X represents the radius r and HIST the concentration (energy density) times $2r\pi$.

The units of X and Z are [\AA], while the units of HIST are [cm^{-3}] in the HIS2 file and [$\text{eV}/\text{\AA}^3$] in the HISEE2 and HISNE2 files, if the dose is specified as a line dose (DOSEUNITS='cm-1') or can be converted to a line dose. Otherwise, the units will be different. For instance, when an area dose is specified and the extent of the implant area in x direction is zero (XINIT(1)=XINIT(2)), the line dose is undefined. The quantity stored in the HIS2 file then has units of [$\text{cm}^{-3}\text{\AA}^{-1}$] and in the HISEE2 and HISNE2 files [$\text{eV}/\text{\AA}^4$]. They may be interpreted as point responses, and actual concentration or energy densities are obtained by integration over $x[\text{\AA}]$. When the default dose of DOSE = $10^{16} \text{ cm}^{-2} = 1 \text{ }^{-2}$ is used, the point responses may be interpreted as the yield times the probability density function (units [10^{-24}\AA^{-2}] and [$\text{eV}/\text{\AA}^2$], respectively).

The HIS2B and HIS2T files contain the spatial distributions of the ejected atoms as a function of the lateral coordinates x and y for backscattered and transmitted atoms, respectively. (In the code snippet given above, Z has to be replaced by Y). The number of species listed, NSPEC, is the same as for the spatial atom distributions inside the target, stored in the HIS2 file. For the first N1+N2 columns, the coordinates x and y refer to the position where the atom leaves the surface boundary layer, which normally is a few Ångströms above the surface. For the last N2 columns, x and y refer to the position where the ejected atoms originate from. The histogram values are the yield times the probability density function (units [Å^{-2}]).

The HISA2B and HISA2T files contain the distributions of the direction cosines dir_x and dir_y of backscattered and transmitted atoms, respectively. Depending on the LRCOIL parameter of the &DAMAGE record, data for NSPEC=N1 or NSPEC=N1+N2 atom species are written to the files. The histogram values are the yield times the probability density function (units 1).

4.5 3-D Histogram Files

Name	File content	flag
HIS3	3-D histogram of stopped atoms	LHIS3
HISEE3	3-D histogram of deposited electronic energy	LHISEE3
HISNE3	3-D histogram of deposited nuclear energy	LHISNE3

The 3-D histogram files are written by

```

CHARACTER*80 TEXT
WRITE (22) TEXT
WRITE (22) NSPEC, NBOXX, NBOXY, NBOXZ
WRITE (22) (X( IBOXX ), IBOXX=1, NBOXX)
WRITE (22) (Y( IBOXY ), IBOXY=1, NBOXY)
WRITE (22) (Z( IBOXZ ), IBOXZ=1, NBOXZ)
DO, I=1, NSPEC
  WRITE (22) ((HIST(I, IBOXX, IBOXY, IBOXZ), &
              IBOXX=1, NBOXX), IBOXY=1, NBOXY), IBOXZ=1, NBOXZ)
ENDDO

```

TEXT is a character string containing information about the stored histogram. NBOXX, NBOXY, and NBOXZ denote the number of histogram boxes in x , y , and z direction, respectively. Regarding adaptation of the histogram during the simulation, the same applies as for the 1-D and 2-D histograms.

X(IBOXX), Y(IBOXY), and Z(IBOXZ) denote the x , y and z coordinate of the center of the histogram box (IBOXX, IBOXY, IBOXZ). HIST(I, IBOXX, IBOXY, IBOXZ) is the concentration of the histogram for species I in the box (IBOXX, IBOXY, IBOXZ).

The units of X, Y and Z are [\AA], while the units of HIST are [cm^{-3}] in the HIS3 file and [$\text{eV}/\text{\AA}^3$] in the HISEE3 and HISNE3 files, if the dose is specified as an ion count (DOSEUNITS='1') or can be converted to an ion count. Otherwise, for instance, when the extent of the implant area in both x and y direction equals zero and an area dose is specified, the units are [$\text{cm}^{-3}\text{\AA}^{-2}$] in the HIS3 file and [$\text{eV}/\text{\AA}^5$] in the HISEE3 and HISNE3 files, so when integrated over $x[\text{\AA}]$ and $y[\text{\AA}]$ one obtains [cm^{-3}] and [$\text{eV}/\text{\AA}^3$], respectively. When the default DOSE = 1 is used, the point response may be interpreted as the yield times the probability density function (units [10^{-24}\AA^{-3}] and [$\text{eV}/\text{\AA}^3$], respectively).

4.6 Miscellaneous Output Files

Name	File content	flag	record
BCK	Backup = data necessary to continue the simulation	NIONBK	&OUTPUT
SE	table of stopping powers	LTAB	&SEPAR
PMAX	maximum impact parameters	LPMAXTAB	&SNPAR
TRA	Information about the trajectories	LTRA, ITRA	&OUTPUT

The backup file BCK contains all information about statistic quantities at the end of the simulation. It may be loaded by another simulation specifying USEBK=T on the &SETUP record. In new simulations, the backup file should only be used for debugging purposes, see the remark towards the end of Section 4.1.

The electronic stopping power file SE lists for each region, ion atom species, and target atom species the electronic energy loss as a function of energy. Each block is written by the following FORTRAN statements:

```
WRITE (122,*) 3,101
WRITE (122,*) 0,0,0
DO, IE=0,100
  WRITE (122,*) E(IE),DEDXNL(IE),DEELOC(IE)
ENDDO
```

E denotes the energy [eV], which is logarithmically distributed between the implant energy (ENERGY parameter of the &IONS record) and the cut-off energy for ion trajectories (EF parameter of the SNPAR record). DEDXNL is the nonlocal energy loss per path length [eV/Å], and DEELOC the local energy loss [eV] at the impact parameter specified by PTAB on the &SEPAR record.

The impact parameter file PMAX is written by the following FORTRAN statements:

```
WRITE (122,*) 4,1000
WRITE (122,*) 0,0,0,0
DO, IE=1,1000
  WRITE (122,*) E(IE),POUT(IE),PNEAR(IE),PIN(IE)
ENDDO
```

E denotes the energy [eV], which is logarithmically distributed between the implant energy ENERGY and the cut-off energy for ion trajectories EF. POUT is the maximum impact parameter used outside the target. PNEAR is the maximum impact parameter used inside the target at distances from the surface smaller than DNEAR (see Section 3.4). PIN is the maximum impact parameter used inside the target at distances from the surface larger than DNEAR. The column for POUT is not written unless PSIFFP2 is set (see Section 5.8). The column for PNEAR is not written if LSPUT=F. If not all columns are written, the first

two lines of the file are modified to reflect the reduced number of columns. Impact parameter tables are written one after the other to the PMAX file for all regions (in case of a non-dynamic simulation) and all atom species.

The trajectory file TRA contains information about the trajectories depending on the variables LTRA and ITRA of the &OUTPUT record. Each line is written by the following FORTRAN statement:

```
WRITE (23,*) X,Y,Z,DIRX,DIRY,DIRZ,E,I1,IG,IFLAG
```

X, Y, and Z denote the coordinates in units of [Å], DIRX, DIRY, DIRZ the direction vector ($\text{DIRX}^2 + \text{DIRY}^2 + \text{DIRZ}^2 = 1$), and E the energy of the projectile in units of [eV]. I1 is the atom index of the projectile, and IG the index of the recoil generation (IG=1 for the ion, IG=2 for the primary recoils, IG=3 for the secondary recoils ...). In case of a virtual recoil (see Section 3.4), I1 is replaced by -I1. IFLAG indicates the position of the point in the trajectory:

IFLAG	Position in the trajectory
0	no trajectory (recoil with subthreshold energy or when LRCOIL=F)
-1, 1	at the start of a trajectory (where a vacancy is generated)
2	along a trajectory
3	at the end of a trajectory inside the target (where an interstitial is generated)
4	at the end of the trajectory of a backscattered ion or recoil
5	at the end of the trajectory of a transmitted ion or recoil

IFLAG=-1 writes the position before lattice vibrations and shifts due to the time integral. IFLAG=1 writes the position after consideration of these two shifts. Note that there is a function in the `read_output.py` script for reading trajectory files (see Section 6.2).

Chapter 5

Input Records

The input language of IMSIL uses the FORTRAN90 syntax for NAMELIST input. Each namelist (here called **input record**) in the input file `INP` contains a set of input parameters. An input record starts with an `&`, followed by the name of the record and the entities (parameters) to be given values. It ends with a slash. Everything outside an input record is ignored and can be used as a comment, typically at the beginning of the file.

While all names are case-insensitive, in this manual parameter and record names are written uppercase to better distinguish them from normal text.

A typical input record reads

```
&name parname1=value1 parname2=value2 ... /
```

`&name` is the name of the NAMELIST record, `parname<i>` is the *i*-th parameter name, and `value<i>` the value to be assigned to `parname<i>`. Values of logical parameters may either be `T` (true) or `F` (false). Character values have to be quoted.

There are three kinds of array parameters in IMSIL: First, **simple arrays** are one-dimensional arrays of fixed size (often the size is small like 2 or 3) that are transparent through the namelist interface. They have to be specified according to the rules for arrays in namelists, i.e. the elements are specified after the `=` sign separated by commas:

```
&name arrayname=value1,value2,value3 ... /
```

Second, **character arrays** are one-dimensional arrays whose dimensions are not fixed at compile time. Their values are specified in a character string, separated by commas¹:

```
&name arrayname='value1,value2,value3' ... /
```

¹The special treatment of this kind of arrays is necessitated by the FORTRAN90 rule saying that allocatable arrays must not appear in NAMELIST statements, and still facilitates namelist input in later versions of FORTRAN.

Third, **index variable arrays** are multi-dimensional arrays or arrays depending on some standard indices like the region or atom index. They are specified by multiple specification of the record with different index variables. In the example of Section 4,

```
&MATERIAL REGION=2 NAME='SI3N4' ... /  
&MATERIAL REGION=1 NAME='SiO2' ... /  
&MATERIAL REGION=3 NAME='SI' ... /
```

the array `NAME(1:3)` is specified by three `&MATERIAL` records with the index variable `REGION`. For records containing index variables, no parameters must be specified that are not arrays of that index. On the other hand, if an array appears on a record that does not contain one or more of its index variables, the values for all those indices are set to the specified value.

As a convenient feature of NAMELIST inputs, only those parameters need to be specified whose predefined default values (given by the program before reading the NAMELIST record) are to be changed. Examples of actual input files are given in Chapter 7.

5.1 *&SETUP*

This record is used to specify some basic parameters of the simulation. It has to appear in each input file. It has no index variables.

ATOM1 — Atom index of histogram file

This variable specifies the index of the atom in the histogram file **FILHIS** whose concentration will be used to initialize the concentration of atoms with index **ATOM2**. Relevant only if **USEHIS=T**.

Type: integer
Default: - (obligatory if **USEHIS=T**)
Range: **ATOM1** > 0

ATOM2 — Atom index of atom species

This variable specifies atom index of atom species which concentration will be initialized with **ATOM1** concentration from histogram file **FILHIS**. Relevant only if **USEHIS=T**.

Type: integer
Default: - (obligatory if **USEHIS=T**)
Range: $1 \leq \text{ATOM2} < \text{NATOM}$

BKFIL — Backup file name

This variable specifies the name of the backup file to be used to continue a previous simulation. Relevant only if **USEBK=T**.

Type: character string
Default: **BCK**
Range: Any name (≤ 80 characters) of an **IMSIL** backup file

EQUICELL — Equidistant cell grid flag

This parameter specifies whether in a 1-D dynamic simulation (LDYN=T) the cell contents are interpolated back to an equidistant grid. Note that this will introduce artificial diffusion. EQUICELL is ignored for a 2-D simulation, since 2-D simulations are always done on equidistant grid. EQUICELL=T can be set to make a 1-D simulation consistent with a 2-D simulation. Otherwise, EQUICELL=F is recommended.

Type: logical

Default: F

Range: T, F

FILCELL — Cell file name

This parameter specifies the name of the file containing the atom densities in the 1-D target cells for dynamic simulation (LDYN=T). The format of the file is described in Chapter 4). Relevant only if USECELL=T.

Type: character string

Default: – (obligatory if USECELL=T)

Range: any name (≤ 80 characters) of an IMSIL cell file

FILHIS — Histogram file name

This variable specifies the name of the histogram file to be used to create a damage profile for subsequent simulation. The format of the histogram file (HIS) is described in file in Chapter 4). Relevant only if USEHIS=T.

Type: character string

Default: – (obligatory if USEHIS=T)

Range: Any name (≤ 80 characters) of an IMSIL histogram file

IARAND — Seed of random number generator

This variable specifies one of the integer numbers defining the seed of the random number generator.

Type: integer

Default: 11113

Range: $20000 < \text{IARAND} < 90000$ (recommended)

IBRAND — Seed of random number generator

This variable specifies one of the integer numbers defining the seed of the random number generator.

Type: integer
Default: 104322
Range: $130000 < \text{IBRAND} < 220000$ (recommended)

IRAND — Seed of random number generator

This variable specifies one of the integer numbers defining the seed of the random number generator. Note that changing only `IRAND` may result in an identical series of random numbers after a relatively small number of numbers.

Type: integer
Default: 4181
Range: $0 < \text{IRAND} < 10000$ (recommended)

LDYN — Dynamic simulation flag

This variable specifies whether the target densities shall be updated in a spatially resolved, cell-based target description. In the dynamic mode the cell densities are updated during the simulation and subsequently used in the selection of collision partners. The cell densities can be output to the `CELL*` files (see Section 4) using `LCELL=T` on the `&OUTPUT` record.

Type: logical
Default: **F**
Range: **T, F**

LSPUT — Sputtering simulation flag

This variable specifies whether accurate sputtering simulation is to be performed. If `LSPUT=T` the surface binding energy is considered. In addition, “virtual” recoils are generated (see Section 3.4). Also the free flight paths in amorphous target are adjusted such that no collision with energy transfer larger than the surface binding energy is missed.

Type: logical
Default: **F**
Range: **T, F**

MRAND — Seed of random number generator

This variable specifies one of the integer numbers defining the seed of the random number generator.

Type: integer
Default: 971
Range: $1000 < \text{MRAND} < 9000$ (recommended)

NATOM — Number of atom species

This variable specifies the number of atom species contained in the ion and the target regions. E.g. for BF_2 implantation in a two-layer structure of SiO_2 and Si, **NATOM**=4 corresponding to the atom species B, F, Si, and O. A target atom is always considered different from an ion atom, even if the same chemical element is specified. E.g., for Si implantation into bare Si **NATOM**=2. This is because by default the mass of an ion atom is set to the mass of the most abundant isotope while the default mass of the target atom is set to the atom mass averaged over all isotopes weighted with their natural abundance. Also, it allows to “label” the ion atoms so they can be distinguished from the target atoms. If the target information is read from a **CELL** or **CELL2** file (**USECELL**=T), **NATOM** depends on whether a dynamic simulation is performed: If **LDYN**=T, **NATOM** must equal the number of atom species stored in the **CELL*** file. If **LDYN**=F, **NATOM** must equal the number of ion atoms (e.g., 2 for BF_2) plus the number of atom species stored in the **CELL*** file.

Type: integer
Default: – (obligatory)
Range: must equal the number of atoms in ion and target

NDIM — Number of dimensions

This variable specifies the number of dimension of the geometry. Depending on **NDIM**, the geometry is defined by the position of the interface (for **NDIM**=1), or by polygons (for **NDIM**=2) defined by corner points (interface points). See **&GEOM** record

Type: integer
Default: 1
Range: 1, 2

NR — Number of regions

This variable specifies the number of target regions. It is ignored when the target geometry is defined by cells (USECELL=T).

Type: integer
Default: - (obligatory if USECELL=F)
Range: NR > 0

RANDSKIP — Number of random numbers to be skipped

This variable specifies the number of random number evaluations before the start of the simulation. Thus, the first RANDSKIP generated numbers of the random number generator are ignored. This is a simpler alternative to specifying IRAND, IARAND, IBRAND, and MRAND. The use of RANDSKIP is problematic for amorphous targets.

Type: integer
Default: 0
Range: RANDSKIP ≥ 0

RNG — Random number generator

This variable specifies the random number generator used to generate uniform random numbers in the range [0,1]. 'Haas' uses the "The multiple prime random number generator" (ACM Trans. Math. Software 13 (4) 368-381, 1987). 'Fortran' uses the default Fortran random number generator RANDOM_NUMBER(). 'Knuth' uses "Knuth's subtractive random number generator" as described in W.H. Press et al., Numerical Recipes, 3rd edition, Cambridge University Press, 2007; p.354. Note that the seed of the random number generator may be set by MRAND, IARAND, IBRAND, and IRAND. The seed does not completely specify the internal state of Knuth's random number generator, but different seeds lead to different random number sequences.

Type: character string
Default: 'Knuth'
Range: 'Haas', 'Fortran', 'Knuth'

REFLECT — Initial condition reflection flag

This variable specifies whether distributions read from HIS or CELL files are reflected at the origin. If SHIFTT0 is also specified, reflection is performed after shifting.

Type: logical
Default: F
Range: T, F

SHIFTT0 — Shifted surface position for initial condition

This variable specifies the position the surface of distributions read from HIS or CELL files should be shifted to. If REFLECT is also specified, reflection is performed after shifting.

Type: real
Default: - (corresponding to no shift)

TEMP — Temperature of the target

This variable specifies the temperature of the target in Kelvin. It is used for the calculation of the vibrational amplitude using the Debye function. See also TDEBYE

Type: real
Default: 300.
Range: TEMP > 0

USEBK — Use-backup-file flag

This variable specifies whether a backup file shall be used to continue a previous simulation. If `USEBK=T`, the backup file specified by `BKFIL` must exist. If `USEION=T`, the input file must specify a larger dose `DOSE` and a larger number of ions `NION` on the `&IONS` record than those stored in the backup file. The continued simulation starts with the ion and damage histograms stored in the backup file and performs a simulation with a number of ions corresponding to the difference of the numbers specified in the input and backup files and a dose corresponding to the difference of the doses specified in the input and backup files. It is not recommended that other parameters than `NION` and `DOSE` be changed in the input file between the previous and the continued run when `USEION=T`. If `USEION=F`, a new simulation is started from scratch except that the damage histograms are read in from the backup file at the beginning of the simulation.

Type: logical
Default: F
Range: T, F

USECELL — Use-cell-file flag

This variable specifies whether the target information shall be read from a cell file as specified by the `FILCELL` parameter. If `USECELL=T`, the cell file specified by `FILCELL` must exist, and the material and geometry specifications in the `&MATERIAL` and `&GEOMETRY` records are ignored. The cell file must be a `CELL` file (see Section 4). If `USECELL=T` and `LDYN=T`, the atomic numbers of the ion atoms stored in the cell file must be consistent with the atom species given in the ion name (`NAME` parameter of the `&IONS` record).

Type: logical
Default: F
Range: T, F

USEHIS — Use-histogram-file flag

This variable specifies whether a histogram shall be used to initialize the damage distribution. If `USEHIS=T`, the histogram file specified by `FILHIS` must exist.

Type: logical
Default: F
Range: T, F

USEION — Use-ion-distribution-from-backup-file flag

This variable specifies whether the ion distribution stored on the backup file shall be used as initial values. If `USEION=F`, only the damage distribution is taken from the backup file and the ion distribution is set to zero. See also `USEBK`.

Type: logical

Default: T

Range: T, F

5.2 &GEOMETRY

The geometry of the target is either described in terms of regions or in terms of cells. In 1-D, the geometry is simply defined by the positions of the surfaces and the interfaces between the regions (POSIF), e.g.,

```
&geometry posif='-10,0,10000' /
```

Cells are usually finer-grained than regions. Only regions are specified on the &GEOMETRY record. Cells are either defined by reading a CELL file, or they are internally generated from regions in case of a dynamic simulation (LDYN=T).

In 2-D, the geometry of each region is specified by a closed polygon. To define the polygons, it is first necessary to specify the coordinates of the vertices (corner points) using the POINT index variable, e.g.,

```
&geometry point=1 pos=0,0 /
&geometry point=2 pos=-5000,0 /
...
```

and then to specify the polygons by listing the point indices in the desired sequence, using the REGION index variable:

```
&geometry region=0 points='1,2,3,4,5,6,7,8' /
&geometry region=1 points='1,4,5,6' /
...
```

Note that the circumference of the complete simulation area has to be specified as region 0. There must not be undefined areas within the circumference, and no regions may overlap.

For 2-D geometries, IMSIL uses an internal mesh containing the minimum distance of each cell from the boundary. It is used to reduce the number of checks that are necessary to determine whether the ion leaves a region.

It is possible to impose cylindrical (LCYLX=T, LCYLY=T, or LCYLZ=T) or spherical (LSPHER=T) symmetry. For cylindrical geometry, the distance along the axis of symmetry replaces the x coordinate, while the distance from the axis replaces the z coordinate. For spherical symmetry, the distance from the center of symmetry replaces the z coordinate.

CENTER — Center of cylindrical and spherical regions

This parameter specifies the center of the cylindrical or spherical regions, if `LCYLX=T` or `LCYLY=T` or `LCYLZ=T` or `LSPHER=T`. The three components of `CENTER` correspond to x , y , and z coordinate of the center. For cylindrical symmetries, the symmetry line passes through the position specified by `CENTER` and is parallel to the x , y , or z axis depending on whether the `LCYLX`, `LCYLY`, or `LCYLZ` flag is set.

Type: simple array (3) of real

Default: 0., 0., 0.

Range: arbitrary

FILGEOM — Geometry file

This parameter specifies the name of a geometry file, which contains further `&GEOMETRY` records with `POINT` and `REGION` definitions. This has the same effect as if the contents of the geometry file was inserted after the last `&GEOMETRY` record of the input file. Any other content of the geometry file other than `&GEOMETRY` records is ignored. If multiple geometry files are specified, only the last one will be included. The geometry file is intended for the specification of 2-D geometries (points and regions), so one geometry file can be used for multiple input files. If a geometry file is specified, the geometry points and regions are not written to the output file. `FILGEOM` is ignored for 1-D geometries (`NDIM=1`).

Type: character string

Default: -

Range: Any valid file name (≤ 80 characters)

FRACUPD — Minimum fractional density change for cell update

A mesh update is triggered in a 1-D dynamic simulation if the change of the cell width relative to the nominal cell width (`wBOX` of the `&OUTPUT` record) would exceed `FRACUPD` in any cell. The cell widths change due to relaxation of density changes. The condition is only checked every `NIONUPD` ions.

Type: integer

Default: 0.1

Range: ≥ 0

LPERX — Periodic x-boundary conditions flag

This parameter specifies whether periodic boundary conditions should be used in x-direction in a 2-D simulation. LPERX=T requires specification of XPER.

Type: logical
Default: F
Range: T, F

LCYLX — Cylinder region flag, axis parallel x

This parameter specifies that the target should be considered cylindrical symmetric. The axis of rotation is parallel to the x axis and goes through CENTER. For NDIM=1 the radii of the cylindrical regions are specified by POSIF. For NDIM=2 the polygon defined by POS and POINTS is rotated around the axis. POS(1) is considered the coordinate along the axis, POS(2) the distance from the axis (radius). Any combination of two or more of LCYLX=T, LCYLY=T, LCYLZ=T, LSPHER=T, and LROBIN=T is illegal. LCYLX=T can only be specified for LDYN=F.

Type: logical
Default: F
Range: T, F

LCYLY — Cylinder region flag, axis parallel y

This parameter specifies that the target should be considered cylindrical symmetric. The axis of rotation is parallel to the x axis and goes through CENTER. For NDIM=1 the radii of the cylindrical regions are specified by POSIF. For NDIM=2 the polygon defined by POS and POINTS is rotated around the axis. POS(1) is considered the coordinate along the axis, POS(2) the distance from the axis (radius). Any combination of two or more of LCYLX=T, LCYLY=T, LCYLZ=T, LSPHER=T, and LROBIN=T is illegal. LCYLY=T can only be specified for LDYN=F.

Type: logical
Default: F
Range: T, F

LCYLZ — Cylinder region flag, axis parallel z

This parameter specifies that the target should be considered cylindrical symmetric. The axis of rotation is parallel to the x axis and goes through **CENTER**. For **NDIM=1** the radii of the cylindrical regions are specified by **POSIF**. For **NDIM=2** the polygon defined by **POS** and **POINTS** is rotated around the axis. **POS(1)** is considered the coordinate along the axis, **POS(2)** the distance from the axis (radius). Any combination of two or more of **LCYLX=T**, **LCYLY=T**, **LCYLZ=T**, **LSPHER=T**, and **LROBIN=T** is illegal. **LCYLZ=T** can only be specified for **LDYN=F**.

Type: logical
 Default: F
 Range: T, F

LSPHER — Spherical region flag

This parameter specifies that the target regions should be considered spherical around the center given by **CENTER** rather than planar. The radii of the spherical regions are specified by **POSIF**. Any combinations of **LCYLX=T**, **LCYLY=T**, **LCYLZ=T**, **LSPHER=T**, and **LROBIN=T** must not be specified. **LSPHER=T** can only be specified for **NDIM=1**, **LDYN=F**.

Type: logical
 Default: F
 Range: T, F

NBOX2 — Maximum number of mesh boxes

This parameter specifies the maximum number of boxes in each direction of the internal mesh. Geometry checks are performed by interpolation of the distance function stored on the internal mesh. Therefore, a finer mesh means a finer resolution of the geometry. The first value of **NBOX2** corresponds to the number of boxes in x -direction, the second value corresponds to the number of boxes in z -direction. Each of the values of **NBOX2** is effective only if the number of boxes necessary to cover the geometry with a box width of the respective value of **WBOX2** would exceed **NBOX2**. In this case the box width is adjusted so that the number of boxes equals **NBOX2**. Thus, **NBOX2** sets upper limits to the number of boxes used in the simulation. The first (x) value of **NBOX2** is ignored if **LPERX=T**.

Type: simple array (2) of integer
 Default: 1000, **NBOX2(1)**
 Range: > 0

NIONUPD — Number of ions for mesh update

This parameter specifies the number of implanted ions after which the cell sizes are updated in a 1-D dynamic simulation, see `FRACUPD`). Note that too infrequent mesh updates affect the simulation results. `NIONUPD` must not appear on `POINT` or `REGION` records.

Type: integer
 Default: 1
 Range: > 0

POINT — Point index

This parameter specifies the index of a point whose coordinates are specified on the same record by `POS`. The point indices are referred to by the `POINTS` arrays.

Type: integer
 Default: –
 Range: > 0

POINTS — Points defining a region polygon

This parameter specifies the indices of the points (as defined by `POINT`) defining the region polygon. The character array may be split to several records. Specification of `POINTS` requires the index variable `REGION` on the same record.

Type: index variable array (`REGION`) of character array
 Default: –
 Range: point indices defined by `POINT`

POS — Position of an interface point

This parameter specifies the position of an interface point. The position is a simple array of size 2, consisting of the x and z coordinate in case of Cartesian coordinates. In case of cylindrical coordinates `POS(1)` refers to the distance along the axis of symmetry, and `POS(2)` to the distance from the axis of symmetry. The units of the coordinates are Å. Specification of `POS` requires the index variable `POINT` on the same record.

Type: index variable array (`POINT`) of simple array (2) of real
 Default: –
 Range: arbitrary

POSFIX — Fixed target position

This parameter specifies the position which remains fixed during target relaxation in a 1-D dynamic simulation (NDIM=1, LDYN=T) or during the initial relaxation when the geometry is read from a CELL file (NDIM=1, USECELL=T). If POSFIX is unspecified, it is assumed in the center of the target. The center position is updated during a dynamic simulation. POSFIX also falls back to the target center if it happens to fall outside the target in the course of a dynamic simulation. The units of POSFIX are Å.

Type: real
 Default: center of simulation domain
 Range: arbitrary

POSIF — Positions of the interfaces

This parameter specifies the positions of the interfaces between neighboring regions in case of a 1-D simulation (NDIM=1 on the &SETUP record. The first value of POSIF defines the front surface of the target, the last value the back surface. The character array may be split to several records. The units of POSIF are Å.

Type: character array (NR+1) of real
 Default: 0., 10^{30} if only one region,
 required otherwise
 Range: values must be increasing

WBOX2 — 2-D box width

This parameter specifies the box width of the internal mesh. The first value corresponds to the cell width in x-direction, the second value corresponds to the cell width in z-direction. The values of WBOX2 might be adjusted by IMSIL if the maximum number of boxes NBOX2 were exceeded, or in case of periodic boundary conditions to fit an integer number of intervals between the boundaries. The units are Å. Geometry checks are performed by interpolation of the distance function stored on the internal mesh. Therefore, a finer mesh means a finer resolution of the geometry.

Type: simple array (2) of real
 Default: 1.0, WBOX2(1)
 Range: > 0

WROUGH — Thickness of roughness layer

This parameter specifies the thickness of the surface roughness layer. Within the roughness layer, the density is assumed to increase linearly from 0 (at the surface) to its nominal value (at the other bound of the roughness layer). The units are Å.

Type: real
Default: 0
Range: ≥ 0

XPER — x-positions of periodic boundaries

This parameter specifies the positions of the boundaries in case of periodic boundary conditions (LPERX=T). No points (POS) must be outside these boundaries. In general, some points will be on the boundaries. These points will be snapped to the boundary with a tolerance corresponding to the accuracy of single precision floating point data. The units are Å.

Type: simple array (2) of real
Default: -
Range: $XPER(2) > XPER(1)$

5.3 &ATOMS

This record is used to define the properties of the atoms. Normally, it is not necessary to include this record, since the atom properties are defined internally, and the atom species are defined on the &IONS and &MATERIAL records or, if the target is initialized from a CELL file, in the CELL file. However, this record can be used to change the default values of the atom properties.

All parameters of this record are index variable arrays with the index variable `ATOM`. `ATOM` is the atom index. The mapping between atom names and atom indices is defined by the sequence of occurrence of the atom names in the ion and material names: First the ion and then the materials from region 1 to how many regions are defined. Within the ion and material names from left to right. Using the same example as in Section 4.3,

```
&IONS NAME='B', ... /
&MATERIAL REGION=2 NAME='SI3N4' ... /
&MATERIAL REGION=1 NAME='SiO2' ... /
&MATERIAL REGION=3 NAME='SI' ... /
```

atom indices and names relate as follows:

ATOM	NAME
1	B
2	SI
3	O
4	N

Note that material names are ordered according to the `REGION` index rather than their appearance in the input file.

Occasionally, the same atom species may occur in the ion and one or more materials. As an example, take

```
&IONS NAME='CN', ... /
&MATERIAL REGION=1 NAME='C5H8O2' ... /
&MATERIAL REGION=2 NAME='C' ... /
```

For a static simulation (`LDYN=F`), the C atom in the ion is treated as separate atom species from the C atoms in the materials:

ATOM	NAME
1	C
2	N
3	C
4	H
5	O

The C atoms in the ion may be assigned different properties than the C atoms in the two materials. On the other hand, in a dynamic simulation (`LDYN=T`), the C atoms in the ion are not distinguished from the C atoms in the materials:

ATOM	NAME
1	C
2	N
3	H
4	O

When a `CELL` file is read rather than regions being defined, the atom incidences are defined by the order as the atomic numbers appear in line 2 of the `CELL` file (cf. Section 4.3). The ion atoms as specified in the `&IONS` record must appear in the `CELL` file first and in the same order as in the `&NAME` parameter. That means, in the current version of IMSIL, the ion atom species in a dynamic simulation must already be in the target when the target is read from a `CELL` file. If necessary, the `CELL` file must be prepared by an external tool (script) to include the new ion species atom(s).

$1 \leq \text{ATOM} \leq \text{NATOM}$ is required, where `NATOM` denotes the number of atom species in the simulation.

DENSITY — Atom density

This parameter specifies the number of target atoms per cm^3 in a pure material.

Type: index variable array (`ATOM`) of real
 Default: depending on `NAME`
 Range: > 0

EFAKE — Energies for fake thermal vibrations

This parameter specifies the ion energies where the pseudo-temperatures `TEMPFAKE` are defined. See `TEMPFAKE` for further information. The units of `EFAKE` are keV.

Type: index variable array (`ATOM`) of simple array (2) of real
 Default: –
 Range: > 0

ESURF — Surface binding energy

This parameter specifies the surface binding energy of the pure material in eV. Default values are at least 0.1 eV, since lower surface binding energies lead to excessive simulation times. The user may explicitly set any positive surface binding energies, though, in the input file.

Type: index variable array (`ATOM`) of real
 Default: from internally tabulated values
 Range: > 0

MASS — Atom mass

This parameter specifies the mass of the atom in units of the atomic mass unit. It overwrites the default values determined by the atom names.

Type: index variable array (**ATOM**) of real
Default: determined from **NAME**
Range: > 0

NAME — Name of atom

This parameter specifies the chemical formula of the atom.

Type: index variable array (**ATOM**) of character(2)
Default: - (obligatory)
Range: any chemical name of an atom

TDEBYE — Debye temperature

This parameter specifies the Debye temperature of the pure material. It is used for calculation of vibrational amplitude using Debye function. **TDEBYE=0** means a thermal vibration amplitude of 0.083 will be used. Note that some internally tabulated values (e.g. for noble gases) are zero. The units of **TDEBYE** are K.

Type: index variable array (**ATOM**) of real
Default: internally tabulated values
Range: > 0

TEMPFAKE — Temperatures for fake thermal vibrations

This parameter specifies the pseudo-temperatures to be used for the calculation of the amplitudes of the fake thermal vibrations. Each value corresponds to an ion energy as specified by the parameter **EFAKE**. If for an atom index **ATOM** no value of **TEMPFAKE** is specified, fake thermal vibrations are disabled. If for an atom index **ATOM** only one value of **TEMPFAKE** is specified, the amplitude of the fake thermal vibrations is constant. If two values are specified, the amplitude of the fake thermal vibrations is assumed to depend on the ion energy E according to a power law such that for $E = \text{EFAKE}(1)$ the amplitude corresponds to a temperature **TEMPFAKE(1)** and for $E = \text{EFAKE}(2)$ the amplitude corresponds to a temperature **TEMPFAKE(2)** according to the Debye model. Thus, if two values of **TEMPFAKE** are given, two values of **EFAKE** must be specified as well. Note that the larger of the physical and fake thermal vibration amplitude is used and fake thermal vibrations are ignored in amorphous materials. The units of **TEMPFAKE** are K.

Type: index variable array (**ATOM**) of simple array (2) of real
 Default: – (obligatory if **EFAKE** is defined)
 Range: > 0

VIB — Thermal vibration amplitude

This parameter specifies the mean square amplitude in Å of vibrating lattice atoms in a given direction. The displacements of the atoms from their lattice sites is Gaussian distributed.

Type: index variable array (**ATOM**) of real
 Default: Calculated from **TDEBYE**
 Range: ≥ 0

5.4 &IONS

This record is used to describe the ions. It has to appear in each input file. It has the index variable **ATOM1** corresponding to the index of the atom in the ion. The index is increasing from left to right in the ion name as specified by the **NAME** parameter. E.g., for **NAME='BF2'**, **ATOM1=1** corresponds to B and **ATOM1=2** to F.

Another index variable is **BEAM**. It allows to specify the beam as a superposition of partial beams. Ions are chosen from each of the partial beams with a probability proportional to **WEIGHT**, see Section 2.1.

DIVERG — Beam divergence

This parameter specifies the beam divergence in degrees. For **MODDIV=1**, the initial ion direction will be distributed isotropically within an elliptical cone of apertures **2*DIVERG(1)** and **2*DIVERG(2)** in *x* and *y* direction of the beam coordinate system. For **MODDIV=2**, the initial ion direction will be Gaussian distributed with standard deviations **DIVERG(1)** and **DIVERG(2)** in *x* and *y* direction of the beam coordinate system. The ion direction will be rotated into the target coordinate system according to **TILT** and **ROTATE**.

Type: index variable array (**BEAM**) of simple arrays (2) of real
 Default: 0, **DIVERG(1)**
 Range: ≥ 0

DOSE — Implantation dose

This parameter specifies the implantation dose in units defined by **DOSEUNITS**. If **DOSEUNITS='cm-2'**, the dose is defined as the number of incident ions per area in the *x-y* plane (not perpendicular to the beam if there is a tilt angle). If **DOSEUNITS='cm-1'**, the dose is defined as the number of incident ions per length in *y* direction. For **DOSEUNITS='1'**, **DOSE** refers to the number of incident ions.

Type: real
 Default: corresponding to 1 ion per \AA^2 , \AA , or 1 ion depending on **DOSEUNITS**
 Range: > 0

DOSEUNITS — Units of dose

This parameter specifies the units of the implantation dose specified by DOSE. See the description of DOSE for the meaning of possible units.

Type: real
 Default: 'cm-2'
 Range: 'cm-2', 'cm-1', '1'

ENERGY — Initial ion energy

This parameter specifies the initial ion energy in keV.

Type: real
 Default: - (obligatory)
 Range: > 0

FILTRA — Trajectory file

This parameter specifies the name of a trajectory file used to set the initial conditions of the ions. The format of the trajectory file is described in file in Chapter 4). Lines with IFLAG=2 are ignored. Lines with IFLAG=1 are used to mark lattice atoms that are initially excluded from being a collision partner. The parameters ENERGY and NION of the &ION record still must be specified, but it is sufficient to use upper bounds to the actual values. Normally, one may want to set LIINIT=T. Relevant only if USETRA=T.

Type: character string
 Default: - (obligatory if USETRA=T)
 Range: Any name (≤ 80 characters) of an IMSIL trajectory file

FWHM — Full width at half maximum

This parameter specifies the full width at half maximum of the blur of the ion beam. A Gaussian distribution function with the given FWHM is superposed to the initial positions of the ions. The two values of FWHM correspond to the x and y direction in the beam coordinate system. The units of FWHM are Å.

Type: simple array (2) of real
 Default: 0., FWHM(1)
 Range: > 0

LIINIT — Internal starting point flag

This parameter specifies whether the starting points of the ion trajectories shall be chosen at the coordinates given by XINIT, YINIT, ZINIT (LIINIT=T), or by a projection of these points to the surface along the incidence direction of the ion (LIINIT=F). For LIINIT=T, when the starting point is inside the target, the first free flight path starts there. When it is outside the target, the point of entrance into the target is searched for in the forward direction only (this makes possible to start trajectories inside a hollow cylinder or sphere). For LIINIT=F the entrance into the target is searched for also in the backward direction.

Type: logical
 Default: F
 Range: T, F

LLINIT — Start as lattice atom flag

This parameter specifies whether in the case of an internal starting point (LIINIT=T) the starting point shall be the position of a lattice atom.

Type: logical
 Default: F
 Range: T, F

LTINIT — Start as target atom flag

This parameter specifies whether the ion shall be considered indistinguishable from a target atom if they have the same atomic number. If LTINIT=t and the ion has the same atomic number as a target atom, replacement collisions are allowed and the ion is counted as damage.

Type: logical
 Default: F
 Range: T, F

MASS — Ion mass

This parameter specifies the mass of the ion atoms in units of the atomic mass unit. It overwrites the default values determined by the atom names.

Type: index variable array (ATOM1) of real
 Default: determined from NAME
 Range: > 0

MODDIV — Beam divergence model

This parameter specifies the distribution function of the ion's incidence direction around the nominal beam direction given by `TILT` and `ROTATE`. `MODDIV=1` means that the actual incidence direction is isotropically distributed within the angle `DIVERG` from the nominal incidence direction. `MODDIV=2` means a Gaussian weighted isotropic distribution with standard deviation `DIVERG`. While the isotropic distribution (`MODDIV=1`) is exact for all values of `DIVERG`, the Gaussian weighted distribution is exact only in the limit of small `DIVERG`. Note that for small `DIVERG` the root mean square value of the angle between actual and nominal incidence angle is $DIVERG/\sqrt{2}$ in case of an isotropical distribution (`MODDIV=1`).

Type: index variable array (`BEAM`) of integer
 Default: 1
 Range: 1, 2

NAME — Name of ion species

This parameter specifies the chemical formula of the ion molecule. E.g. `NAME='B'` for B implantation, `NAME='BF2'` for BF_2 implantation, etc. The number of atom species (2 in the case of BF_2), the atom names (B and F), the atomic numbers (5 and 9, respectively), default values of the atom masses (11.009 and 18.998, respectively), and the number of atoms of each atom species in the molecule (1 and 2, respectively) are extracted automatically.

Type: character(80)
 Default: – (obligatory)
 Range: any chemical name of a molecule

NION — Number of simulated ions

This parameter specifies the number of ions to be simulated. The simulation time will be roughly proportional to `NION` while statistical errors decrease with the square root of `NION`. In case of the continuation of a previous run (`USEBK=T`), `NION` specifies the total number of ions including those simulated in the previous run.

Type: integer
 Default: – (obligatory)
 Range: > 0

RANROT — Random rotation angle flag

This parameter is a flag indicating that the rotation angle has to be selected randomly, simulating a continuously varying rotation angle. If RANROT=T, ROTATE must not be specified at the same time.

Type: logical
 Default: F
 Range: T, F

REENTER — Reenter flag

This parameter specifies whether ions are allowed to traverse vacuum regions and to reenter the target somewhere else. If REENTER=F, the ion trajectories are terminated as soon as they leave the target, and the ions are counted as backscattered or transmitted depending on their direction of motion. REENTER=T must not be specified in a 1-D layer simulation (NDIM=1, LCYL=F, LSPHER=T).

Type: logical
 Default: NDIM>1.OR.LCYL.OR.LSPHER
 Range: T, F

ROTATE — Rotation angle

This parameter specifies the rotation angle in degrees, i.e. the azimuthal orientation of the initial ion direction with respect to the the surface vector VSURF (see &MATERIAL record).

Type: real
 Default: 0
 Range: arbitrary

TILT — Tilt angle

This parameter specifies the tilt angle in degrees, i.e. the angle between the initial ion direction and the wafer normal (defined by WAFER in the &MATERIAL record).

Type: real
 Default: 0
 Range: $0 \leq \text{TILT} < 90$

USETRA — Use-trajectory-file flag

This variable specifies whether the ions are initialized with data from a trajectory file. If **USETRA=T**, the trajectory file specified by **FILTRA** must exist.

Type: logical
Default: **F**
Range: **T, F**

WEIGHT — Beam weight

This parameter specifies the weight of each partial beam. Ions will be chosen from the partial beams with probability proportional to **WEIGHT**. The weights will internally be normalized to unity, i.e., as specified they need not sum up to unity.

Type: index variable array (**BEAM**) of real
Default: 1
Range: ≥ 0

XINIT — Initial x value

This parameter specifies the x coordinate in Å of the starting point of the simulated ion trajectories (**LIINIT=T**) or of the focal point of the beam (**LIINIT=F**). If **LIINIT=F**, the starting point is obtained by projection of the focal point to slightly outside the surface. The starting point or focal point is slightly modified in a crystalline target. If two values are given for **XINIT**, the starting or focal point is generated randomly with uniform distribution between the two values.

Type: simple array (2) of real
Default: 0, **XINIT(1)**
Range: **XINIT(2) \geq XINIT(1)**

YINIT — Initial y value

This parameter specifies the y coordinate in Å of the starting point of the simulated ion trajectories (LIINIT=T) or of the focal point of the beam (LIINIT=F). If LIINIT=F, the starting point is obtained by projection of the focal point to slightly outside the surface. The starting point or focal point is slightly modified in a crystalline target. If two values are given for YINIT, the starting or focal point is generated randomly with uniform distribution between the two values.

Type: simple array (2) of real

Default: 0, YINIT(1)

Range: YINIT(2) \geq YINIT(1)

ZINIT — Initial z value

This parameter specifies the z coordinate in Å of the starting point of the simulated ion trajectories (LIINIT=T) or of the focal point of the beam (LIINIT=F). If LIINIT=F, the starting point is obtained by projection of the focal point to slightly outside the surface. The starting point or focal point is slightly modified in a crystalline target. If two values are given for ZINIT, the starting or focal point is generated randomly with uniform distribution between the two values.

Type: simple array (2) of real

Default: 0, ZINIT(1)

Range: ZINIT(2) \geq ZINIT(1)

5.5 *&CRYSTAL*

This record is used to set the wafer orientation and the parameters for the search procedure of collision partners in the crystalline region.

FFOLDM — Maximum flight path to record old lattice sites

This parameter specifies the flight path within which lattice sites may be recorded to be excluded from the list of potential collision partners. Units are Å.

Type: real
Default: 0.5
Range: > 0

FFPMAX — Maximum free flight path

This parameter specifies the maximum free flight path between collisions. The purpose of the maximum free flight path is to avoid skipping of scattering events due to the finite range of the lattice site list. The units are a quarter of the lattice constant for the default diamond lattice, and the minimum of the lattice constants otherwise.

Type: real
Default: 1.0 for default diamond lattice
0.25 otherwise
Range: ≥ 0

LVIB2 — 2-D lattice vibrations flag

This parameter specifies whether the lattice vibrations are assumed 2-D within the plane perpendicular to the ion's direction of motion, or 3-D. LVIB2=T reduces the execution time.

Type: logical
Default: F
Range: T, F

NAME — Name of crystal lattice

This parameter specifies the name of the crystal lattice and thus the crystal lattice system. Note that the hexagonal lattice structures use orthogonal non-primitive unit cells, the edges of which are reported in the output file. Among the diamond lattices the optimized version (' ') is the most efficient implementation. For the other lattices the alternative names all refer to the same implementation. Alternatively, **NAME** may specify the name of a file containing the unit cell definition (see Chapter 4 for details).

Type: character(80)
 Default: ' ' ... (optimized) diamond
 Range: ' ', 'diamond', 'diamond2' ... diamond
 'sc' ... simple cubic
 'bcc' ... body centered cubic
 'fcc' ... face centered cubic
 'zincblende', '3C' ... zincblende
 'wurtzite', 'wurzite', '2H' ... wurtzite
 '4H'
 '6H'
 Any name (≤ 80 characters) of an IMSIL unit cell file

NLOLDM — Maximum number of old lattice sites

This parameter specifies the maximum number of old lattice sites to be excluded from the list of potential collision partners.

Type: integer
 Default: 6
 Range: $0 \leq \text{NLOLDM} \leq 20$

NRAD2 — Search radius squared

NRAD2+1 is the square of the radius of a sphere around the ion position which is assumed to contain the lattice sites of all potential collision partners. **NRAD2** may only be specified for the default diamond lattice. The units are a quarter lattice constant.

Type: integer
 Default: $\text{PMAX}^2 + \text{FFPMAX}^2$
 Range: $0 \leq \text{NRAD2} \leq 32$

PMAX — Maximum impact parameter

This parameter specifies the maximum impact parameter in the crystalline layer. The units are a quarter of the lattice constant for the default diamond lattice, and the minimum of the lattice constants otherwise.

Type: real
 Default: for default diamond lattice:
 $(\pi L \text{ DENS})^{-1/2}$ if NRAD2 is undefined, with
 $L = 0.43 \text{ DENS}^{-1/3}$ for $E > 10 \text{ keV}$
 $L = (0.3 + 0.7 E0/10 \text{ keV}) 0.43 \text{ DENS}^{-1/3}$ for $E < 10 \text{ keV}$;
 $[\text{NRAD2} + 1 - \text{FFPMAX}^2]^{1/2}$ otherwise
 for NAME='diamond': 0.43
 for other lattice structures: - (obligatory)
 Range: > 0

PSOLDM — Maximum scattering angle to record old lattice sites

Lattice sites are excluded from the sites of potential collision partners only if no collision with a scattering angle larger than PSOLDM has occurred meanwhile. PSOLDM has to be specified in degrees.

Type: real
 Default: 30°
 Range: ≥ 0

5.6 &MATERIAL

This record is used to describe the materials of the regions. Special properties of the crystalline region are specified in the &XTAL record (see Section 5.5). The geometry of the target regions is specified in the &GEOMETRY record (see Section 5.2).

The &MATERIAL record has the index variables REGION, ATOM1, and ATOM2, of which REGION is required on each &MATERIAL record. At least one &MATERIAL record has to appear for each region. ATOM1 is the atom index of a projectile, ATOM2 the atom index of a target atom (recoil). Therefore, it is possible to define the properties of an atom as a projectile and as a recoil separately. Regarding the definition of atom indices, see Section 5.3.

DENSITY — Atom density

This parameter specifies the number of target atoms (irrespective of their species) per cm^3 . E.g., DENS=7.E22 for NAME='SI02' means that there are $2.33 \cdot 10^{22} \text{ cm}^{-3}$ silicon atoms and $4.67 \cdot 10^{22} \text{ cm}^{-3}$ oxygen atoms.

Type: index variable array (REGION) of real
 Default: depending on NAME, weighted densities of the elementary targets
 as specified on the &ATOMS record
 Range: > 0

NAME — Material name

This parameter specifies the chemical name of the target material. E.g. NAME='SI' for silicon, NAME='SI02' for SiO_2 etc. The number of atom species, the atom names, their atomic number, and the probability to encounter each species in the target are extracted automatically. Moreover, default values of the atom masses MASS and material densities DENSITY are provided by the program, if NAME is defined.

Type: index variable array (REGION) of character(80)
 Default: – (obligatory)
 Range: any chemical name of a material

LPOLY — Polycrystalline flag

This parameter specifies whether the orientation of the crystal system shall be rotated randomly before each ion. LPOLY=T requires XTAL='yes' and causes WAFER and VSURF to be ignored.

Type: index variable array (REGION) of logical
 Default: F
 Range: T, F

ROTX — Rotation angle of the x axis

This parameter specifies the rotation angle of the x axis with respect to the VSURF vector. The x axis will be along the surface. The x axis together with the surface normal defines the plane to which the results are projected in 2-D simulations. For a line implantation, ions are implanted between XINIT(1) and XINIT(2) (see &IONS record) which are measured along the x axis. This means that the mask edge orientation of the corresponding impenetrable mask is perpendicular to the x axis. Notice that the rotation angle ROTATE (see &IONS record) for the ion incidence direction is not measured with respect to the x axis, but with respect to the surface vector VSURF. The units of ROTX are degrees. ROTX must only be specified for a crystalline region.

Type: index variable array (REGION) of real
 Default: 0
 Range: arbitrary

SYIELD — Sputtering yield

This parameter specifies the sputtering yield, i.e. the ratio of sputtered particles to implanted ions. The sputtering yield is used to calculate the movement of the surface depending on the implanted dose. SYIELD is only allowed for static 1-D simulations when regions are defined (LDYN=F, NDIM=1, USECELL=F).

Type: index variable array (REGION,ATOM1) of real
 Default: 0
 Range: ≥ 0

TDEBYE — Debye temperature

This parameter specifies the Debye temperature of the material. It is used for calculation of vibrational amplitude using Debye function. TDEBYE=0 means a thermal vibration amplitude of 0.083 will be used. Note that some internally tabulated values (e.g. for noble gases) are inaccurate. The units of TDEBYE are K.

Type: index variable array (REGION,ATOM2) of real
 Default: TDEBYE of the &ATOMS record
 Range: > 0

VIB — Thermal vibration amplitude

This parameter specifies the mean square amplitude in Å of vibrating lattice atoms in a given direction. The displacements of the atoms from their lattice sites is Gaussian distributed.

Type: index variable array (REGION,ATOM2) of real
 Default: Calculated from TDEBYE
 Range: ≥ 0

VSURF — Surface vector

This parameter specifies a reference vector in the wafer surface. If ROTX=0, VSURF specifies the direction of the x axis of the target coordinate system in the crystal coordinate system (the x and y axes of the target coordinate system are parallel to the wafer surface, the z axis is perpendicular to the wafer surface). Otherwise, the x axis will be rotated by ROTX with respect to VSURF. VSURF is the reference direction for the rotation angle of the ion incidence direction (see ROTATE in the &IONS record). The components of VSURF have to be given in the crystal coordinate system, i. e. the coordinate system connected with the [100], [010], and [001] axis of the crystal. VSURF is ignored for non-crystalline regions.

Type: index variable array (REGION) of simple array (3) of real
 Default: 1.,1.,0.
 Range: perpendicular to WAFER

WAFER — Wafer surface normal vector

This parameter specifies a vector normal to the wafer surface. The components of **WAFER** have to be given in the crystal coordinate system, i. e. the coordinate system connected with the [100], [010], and [001] axis of the crystal. E. g. for a (110) wafer, a surface normal may be (0,1,1), thus **WAFER**=0,1,1. **WAFER** is ignored for non-crystalline regions.

Type: index variable array (**REGION**) of simple array (3) of real
Default: 0.,0.,1.
Range: perpendicular to **VSURF**

XTAL — Crystalline region flag

This parameter specifies for each region **IR** whether it is amorphous (**XTAL**='no' or **XTAL**='off') or crystalline (**XTAL**='yes' or any other value). Notice that, due to present limitations of the program, only one region may be crystalline. See **XTAL** record for further parameters defining the crystal structure and controlling simulation in crystalline regions. A crystalline region must not be specified in a dynamic simulation (**LDYN**=T).

Type: index variable array (**REGION**) of character(80)
Default: 'no'
Range: any character string

5.7 &DAMAGE

This record is used to describe damage related parameters. For a description of many of the parameters see Section 3.3. The &DAMAGE record has the index variables **REGION**, **ATOM1**, and **ATOM2**. **ATOM1** is the atom index of a projectile, **ATOM2** the index of the target atom (recoil). Regarding the definition of atom indices, see Section 5.3.

DAMAMO — Amorphization damage concentration

This parameter specifies the threshold damage concentration for amorphization of the crystalline layer. If the damage concentration exceeds **DAMAMO** locally, every collision is treated as in an amorphous target. The units are cm^{-3} .

Type: real
 Default: 2.7×10^{22} in Si unless target is read from a **CELL** file, **DENSITY** otherwise
 Range: $\text{DAMAMO} \geq 0$

DAMSAT — Damage saturation concentration

This parameter specifies the damage saturation concentration. The damage concentrations may not exceed **DAMSAT** unless $\text{DAMAMO} < \text{DAMSAT}$. The units are cm^{-3} .

Type: real
 Default: ∞
 Range: > 0

DENSITY — Reference atom density

This parameter specifies the number of target atoms (irrespective of their species) per cm^3 to be used as a reference density for damage saturation and amorphization. This means that in a monatomic target the interstitial and vacancy concentrations will asymptotically approach **DAMSAT** if the density equals **DENSITY**. Likewise, if the interstitial or vacancy density exceeds **DAMAMO**, it will be set to **DENSITY**.

Type: real
 Default: density of the crystalline region if defined
 density of the only region if only one region is defined
 density of silicon, otherwise
 Range: > 0

DIBIIA — interval of energy deposition for IBIIA effect

This parameter specifies the size of the interval around an amorphous/crystalline interface which the energy deposition is averaged over. The units are Å.

Type: real
Default: 10
Range: > 0

DNEAR — maximum near distance from the surface

This parameter specifies the maximum distance from the surface where recoils are followed down to energies lower than **EF** when **LSPUT=T**. For distances from the surface less than **DNEAR/2**, they are followed down to **ESURF**. For distances between **DNEAR/2** and **DNEAR** the cutoff energy is interpolated linearly. Also, within **DNEAR** from the surface virtual recoils are generated (see Section 3.4). Smaller values of **DNEAR** speed up the simulation, but too small values artificially reduce the sputtering yield if **ESURF** < **EF**. The units are Å.

Type: real
Default: 30
Range: ≥ 0

ED — Displacement energy

This parameter specifies the displacement energy of atom **ATOM1** in pure material **ATOM2**. The displacement energy of **ATOM1** in each region will be calculated by linear superposition with weights according to the abundance of each atom in the region. The units are eV.

Type: index variable array (**ATOM2,ATOM1**) of real
Default: 15
Range: > 0

ELIMKP — Energy limit for Kinchin Pease model

This parameter specifies the energy limit in eV below which the recoil trajectories are terminated and damage is calculated according to the modified Kinchin Pease model. As a cutoff energy, **ELIMKP** operates in addition to **EF**, i.e., trajectories are terminated if their energy is either less than **ELIMKP** or **EF**. The default value of 0 thus means that recoils are followed down to **EF**. **ELIMKP** is not applied to externally started ions. It is set to zero if **LCAP=T** or **LPOCK=T**.

Type: real
 Default: 0
 Range: ≥ 0

EMELT — Heat of melting

This parameter specifies the energy per target atom in eV needed to melt the target. **EMELT** is only effective if **LPOCK=T**. The actual value to be used will usually be larger than the theoretical one since some energy will dissipate without generating a molten zone.

Type: real
 Default: 0.52
 Range: > 0

ESURF — Surface binding energy

This parameter specifies the surface binding energy of atom **ATOM1** in pure material **ATOM2**. The surface binding energy of **ATOM1** in each region will be calculated by linear superposition with weights according to the abundance of each atom in the region. The units are eV.

Type: index variable array (**ATOM2,ATOM1**) of real
 Default: **ESURF(ATOM1,ATOM1)**: heat of sublimation
 ESURF(ATOM1,ATOM2) = mean value of **ESURF(ATOM1,ATOM1)** and
 ESURF(ATOM2,ATOM2)
 Range: ≥ 0

FPOCK — Factor for amorphous pocket recrystallization

This parameter specifies a factor used to calculate the reduction in amorphous pocket size due to athermal recrystallization during quenching of the cascade. It is related to a shrinking radius Δr according to $FPOCK = (4\pi DENS(IR)/3)^{1/3} \Delta r$, if it is assumed that the amorphous pocket is spherical and shrinks by Δr independent of the pocket size. FPOCK is only effective if LPOCK=T.

Type: real
 Default: 1
 Range: ≥ 0

FRACRG — Fraction of FREC-1 generated during recoil generation

This parameter specifies the fraction of the additional Frenkel pairs generated due to FREC at the position of recoil generation with respect to the additional Frenkel pairs generated at both the position of recoil generation and at the position where the recoil comes to rest.

Type: real
 Default: 1
 Range: $0 \leq FRACRG \leq 1$

FREC — Recombination factor within recoil cascade

This parameter specifies a factor to be applied to the generated damage concentrations. It is the ratio of the number of stable defects to the number of defects calculated with the displacement energy before any recombination with defects from previous ions is taken into account. Is set to unity if LCAP=T or LPOCK=T.

Type: real
 Default: SPEC='B': 0.3
 SPEC='BF2': 1.2
 SPEC='SI', 'P': 0.8
 SPEC='AS': 2.
 otherwise: $0.0264 \times M_{Ion}$
 Range: > 0

KSURF — Nonplanarity parameter of surface potential

This parameter specifies the nonplanarity parameter of the surface potential. $KSURF=0$ corresponds to a planar surface potential, $KSURF=1$ to an isotropic potential. $KSURF<1$ means refraction away from the surface normal, $KSURF>1$ towards the surface normal. $KSURF>0$ corresponds to refraction at an ellipsoidal surface, while $KSURF<0$ corresponds to refraction at a hyperboloidal surface. For projectiles entering the target, currently any value other than $KSURF=0$ is treated like $KSURF=1$.

Type: real
Default: 0.
Range: any real number

LAM02 — 2-D amorphization flag

This parameter specifies how the positions of “amorphous” displaced atoms are calculated. If $LAM02=T$ an amorphous displaced atom is generated from a regular lattice atom by displacing it perpendicularly to the ion trajectory with a uniform probability distribution within the radius P_{MAX} of the `&CRYSTAL` record from the undeflected ion trajectory. If $LAM02=F$ amorphous displaced atoms are generated at random positions uniformly distributed in the lattice.

Type: logical
Default: T
Range: T, F

LAMZON — amorphous zone model flag

This parameter specifies whether the displaced atoms are combined into amorphous zones of the size of a histogram box ($LAMZON=T$), or considered individually ($LAMZON=F$). If the amorphous zone model is used, the probability of the occurrence of an amorphous zone is chosen such that displaced atoms are encountered with the same overall probability as in the $LAMZON=F$ case.

Type: logical
Default: F
Range: T, F

LCAP — Capture radius flag

This parameter specifies whether damage recombination within a recoil cascade is performed by recording the positions of all vacancies and interstitials and recombining those which are closer to each other than the capture radius RCAP.

Type: logical
Default: F
Range: T, F

LDAM — Damage flag

This parameter specifies whether a refined damage model is used, and whether damage is considered in the crystalline layer (if any) for the trajectories of subsequent ions and recoils. For LDAM=F and LRCOIL=F no damage is recorded. For LDAM=F and LRCOIL=T interstitial and vacancy distributions are recorded, where one interstitial is generated when and where a projectile stops and one vacancy is generated when and where a target atom is recoiled. For LDAM=T and LRCOIL=F the amount of damage is calculated according to the modified Kinchin Pease model, it can be considered as the density of displaced atoms or Frenkel pairs. For LDAM=T and LRCOIL=T one of the refined damage models is used: Default is the model described in Section 3.3, while when LCAP=T, the capture radius model is invoked. For LDAM=T, damage is considered in the selection of collision partners in a crystalline layer. Note that it is not possible to use one of the refined damage models without considering damage for subsequent trajectories in a crystalline layer. If this is desired, use a very small implantation dose (DOSE of the &ION record). Note that moments and histograms of the damage distributions are only output if the appropriate flags are set on the &OUTPUT record. Nuclear energy deposition is calculated independent of LDAM.

Type: logical
Default: .NOT.LDYN
Range: T, F

LIBIIA — IBIIA flag

This parameter specifies whether amorphous zones may grow according ion beam induced interfacial amorphization (IBIIA).

Type: logical
Default: F
Range: T, F

LPOCK — Amorphous pocket flag

This parameter specifies whether amorphous pockets are formed based on the parameters EMELT and EPOCK. Currently amorphous pockets are not used in subsequent trajectory simulations, but have only the effect of increased damage. Currently LPOCK=T must not be specified at the same time as LVAC=T.

Type: logical

Default: F

Range: T, F

LRCOIL — Recoil cascade flag

This parameter specifies whether the recoil trajectories are to be simulated. Is set true if LCAP=T or LPOCK=T. LRCOIL also determines how damage is calculated (see LDAM).

Type: logical

Default: F

Range: T, F

LREPL — Replacement collision flag

This parameter specifies whether replacement collisions are allowed. Replacement collision are collisions where the projectile stops and the recoil moves on, provided projectile and recoil are the same species. In a replacement collision no defects are generated, while if the same collision is treated without replacement, a Frenkel pair is created.

Type: logical

Default: T

Range: T, F

NDIM — Dimensionality of damage histogram

This parameter specifies the dimensionality of the histogram from which the damage concentration is read for consideration in the crystalline layer (if defined) for the trajectories of subsequent ions and recoils. Note that not all dimensionalities are possible depending on the choices of DOSEUNITS and XINIT and YINIT of the &IONS record (see Section XXX). The default values represent the largest values possible. It is necessary to demand output of the respective damage histogram by setting LHis=T, LHis2=T, or LHis3=T on the &OUTPUT record, depending on NDIM.

Type: integer
 Default: 1 for DOSEUNITS='cm-2' and XINIT(1)=XINIT(2),
 2 for DOSEUNITS='cm-2' and
 XINIT(1)>XINIT(2) and YINIT(1)=YINIT(2),
 or DOSEUNITS='cm-1' and YINIT(1)=YINIT(2),
 3 otherwise
 Range: $1 \leq \text{NDIM} \leq 3$ and not larger than the default values

NPOCKMIN — Minimum number of atoms in pocket

This parameter specifies the minimum number of atoms in an amorphous pocket. If the energy of a tentative amorphous pocket is such that less than NPOCKMIN atoms are contained after quenching, separated point defects are assumed to be generated instead. NPOCKMIN is only effective if LPOCK=T.

Type: integer
 Default: 10
 Range: ≥ 0

PSI110 — Critical angle around $\langle 110 \rangle$

This parameter specifies a critical angle with respect to $\langle 110 \rangle$ directions for application of the modified Kinchin-Pease model. The modified Kinchin-Pease model is not applied if the angle between the direction of motion and a $\langle 110 \rangle$ direction is less than PSI110. The units of PSI110 are degrees.

Type: real
 Default: 0
 Range: $0 \leq \text{PSI110} \leq 180$

RCAP — Capture radius

This parameter specifies the capture radius to be used if `LCAP=T` or `LFLOW=T`. The units of `RCAP` are Å.

Type: real
 Default: $(3/8\pi\text{DAMSAT})^{1/3}$
 Range: > 0

REENTER — Reenter flag for recoils

This parameter specifies whether recoils are allowed to traverse vacuum regions and to reenter the target somewhere else. If `REENTER=F`, the recoil trajectories are terminated as soon as they leave the target, and the recoils are counted as backscattered or transmitted depending on their direction of motion. `REENTER=T` must not be specified in a 1-D layer simulation (`NDIM=1`, `LCYL=F`, `LSPHER=T`).

Type: logical
 Default: `NDIM>1.OR.LCYL.OR.LSPHER`
 Range: T, F

RIBIIA — IBIIA growth rate constant

This parameter specifies the displacement of the amorphous/crystalline interface per deposited nuclear energy density for the IBIIA effect. Its units are Å/(eV/Å³).

Type: real
 Default: 100
 Range: > 0

ZMAXKP — Maximum depth for Kinchin Pease model

This parameter specifies a maximum depth at which the modified Kinchin Pease model is invoked. For depth larger than `ZMAXKP` the Kinchin Pease model is not applied. The units of `PSI110` are degrees.

Type: real
 Default: ∞
 Range: ≥ 0

5.8 *&SNPAR*

This record is used to define the parameters of nuclear stopping. For a description of some of the parameters refer to Chapters 3.1 and 3.2. The *&SNPAR* record has the index variables *ATOM1* and *ATOM2*. *ATOM1* corresponds to the index of the projectile, while *ATOM2* corresponds to the index of the target atom/recoil. Regarding the definition of atom indices, see Section 5.3.

CHKFFPCYL — Check ffp cylinder flag

This parameter specifies whether a check is to be performed on whether a recoil partner in an amorphous region is inside the previous free flight path cylinder. If this is the case, the collision is abandoned.

Type: logical
Default: F
Range: T, F

COEFFIL — Screening function coefficient file

This parameter specifies the file containing the coefficients of a sum-of-exponentials screening function. This model is used for the calculation of the scattering angles, if *QUAD* specifies a numerical integration method and *SCRFUN*='coeffil'. Files are assumed to be located in the subdirectory '*screencoefs*' of the search path.

Type: index variable array (*ATOM2,ATOM1*) of character(80)
Default: '*ZBLspec*'
Range: '*ZBLspec*' or the name of a user-supplied file containing screening function coefficients

DENFFP — Minimum energy transfer

This parameter specifies the minimum energy transfer in eV which has to be considered in a collision. It is used for the calculation of the free flight path in amorphous targets. The free flight path is chosen such that both the energy transfer is larger than *DENFFP* and the scattering angle is larger than *PSIFFP*.

Type: real
Default: 15
Range: > 0

EF — Final projectile energy

This parameter specifies the cutoff energy in eV below which the simulation of the ion trajectory is terminated. For `LSPUT=T` the cutoff energy is reduced to the minimum surface binding energy for position closer to the surface than `DNEAR` of the `DAMAGE` record. Note that additional displacement is possible at the end of the trajectory if `LFLOW=T`. The units of `EF` are eV.

Type: real
 Default: 10
 Range: $0 < EF \leq ENERGY$

ETIME — Maximum energy for time integral

This parameter specifies the energy below which the time integral is considered for the calculation of the turning points of the trajectories. The turning point is the intersection of the asymptotes of the classical scattering problem. If the turning point is beyond the plane perpendicular to the incidence direction that contains the collision partner, the collision point is moved into that plane. The units of `ETIME` are keV.

Type: real
 Default: 5
 Range: ≥ 0

FFPMAX — Maximum free flight path

This parameter specifies an upper limit to the free flight path in amorphous material. `FFPMAX` is applied to the statistical free flight path, not to its mean value obtained from `DENFFP` and `PSIFFP`. `FFPMAX` is silently reduced to the size of the simulation domain in z direction (but not to less than 0.1\AA) in a dynamic simulation (`LDYN=T`) or when the target is read from a `CELL` file (`LCELL=T`). The units of `FFPMAX` are \AA .

Type: real
 Default: 10% of target thickness
 Range: $0 \leq FFPMAX \leq \text{size of simulation domain}$

LAPSISEF — Apsis cutoff energy flag

If true, this parameter specifies that the kinetic energy at the apsis of collision rather than the energy after collision should be checked against **EF** to determine trajectory cutoff. This can provide a better comparison with MD simulations.

Type: logical
Default: **F**
Range: **T, F**

LINEL — Inelastic scattering flag

This parameter specifies whether the local electronic energy loss is considered in the calculation of the scattering angles and energy transfer.

Type: logical
Default: **F**
Range: **T, F**

LNEGX1 — Enforce negative time integral shift flag

This flag, when set, enforces negative (or zero) time integral shifts along the incoming asymptote. This means that in the rare case when the time integral shift is positive, it is set to zero. Only relevant when the time integral is considered, i.e., when the projectile energy is smaller than **ETIME**. For channeling simulations **LNEGX1=T** is recommended, otherwise nearly simultaneous collision may be missed.

Type: logical
Default: **T**
Range: **T, F**

LPMAXTAB — Maximum impact parameter table flag

This parameter specifies whether the maximum impact parameter as a function of energy should be written to the **PMAX** file. The table lists the energy and the corresponding impact parameters in different regions (in terms of distance from the surface).

Type: logical
Default: **F**
Range: **T, F**

LPOSFFP — Enforce positive free flight paths flag

This flag, when set, enforces positive (or zero) free flight paths. Negative free flight paths may occur when LPOSFFP=F if the time integral is considered. For channeling simulations LPOSFFP=F is considered, since this will more reliably catch nearly simultaneous collisions (double scattering is avoided by NLOLDM of the &XTAL record).

Type: logical
Default: F
Range: T, F

LSTFFP — Statistical free flight path flag

This parameter specifies whether the free flight paths are calculated statistically (LSTFFP=T) or deterministically (LSTFFP=F). In both cases, a free flight path is calculated from the parameters DENFFP, FFPMAX, PMAXMAX, PMAXMIN, PMAXPSI, PSIFFP, and the current projectile energy and species. If LSTFFP=F, this free flight path is taken as is. If LSTFFP=T, this value is taken as the mean value of an exponential distribution of the free flight path.

Type: logical
Default: T
Range: T, F

NABSC — Number of abscissae for quadrature of the scattering angle

This parameter specifies the number of abscissas to be used in the quadrature for the calculation of the scattering angle in the center-of-mass system (if QUAD specifies an integration method).

Type: index variable array (ATOM2,ATOM1) of integer
Default: 4
Range: > 0

PMAxmax — Maximum maximum impact parameter in amorphous material

This parameter specifies an upper limit to the maximum impact parameter in amorphous regions (regions with `XTAL='no'`). The actual maximum impact parameter is usually smaller according to `DENFFP` and `PSIFFP`. There may be another effective upper limit to the maximum impact parameter imposed by the scattering table used. The units of `PMAxmax` are Å.

Type: real
 Default: 4
 Range: > 0.

PMAxmin — Minimum maximum impact parameter in amorphous material

This parameter specifies a lower limit to the maximum impact parameter in amorphous regions (regions with `XTAL='no'`). The actual maximum impact parameter is usually larger according to `DENFFP` and `PSIFFP`. The units of `PMAxmin` are Å.

Type: real
 Default: 0.004
 Range: > 0.

PMAxpsi — Maximum maximum impact parameter for the angle criterion

This parameter specifies an upper limit to the maximum impact parameter as determined from the scattering angle criterion (`PSIFFP`). If the scattering angle criterion gives a maximum impact parameter larger than `PMAxpsi`, it is reset to `PMAxpsi`. Note that the maximum impact parameter can be larger than `PMAxpsi` due to `DENFFP`. The units of `PMAxpsi` are Å.

Type: real
 Default: $1.53 / \min(1, \sqrt{0.15 + 0.85 \text{ ENERGY}/10 \text{ keV}})$
 Range: > 0.

PSIF — Final ion angle

This parameter specifies the cutoff angle of the ion in degrees, at which the simulation of the trajectory is terminated. It is measured with respect to the surface normal unless LPINIT=T on the &IONS record, in which case it is measured with respect to the plane defined by the surface normal and ROTATE of the &IONS record. PSIF=0 means that no cutoff angle has to be taken into account.

Type: real
 Default: 0
 Range: $0 \leq \text{PSIF} \leq 90$

PSIFFP — Minimum scattering angle

This parameter specifies the minimum scattering angle in degrees which has to be considered in a collision. It is used for the calculation of the free flight path in amorphous targets. It is chosen such that both the energy transfer is larger than DENFFP and the scattering angle is larger than PSIFFP.

Type: real
 Default: 1
 Range: $0 < \text{PSIFFP} < 180$

QUAD — Method of scattering angle quadrature

This parameter specifies the method of numerical quadrature for the calculation of the scattering angles in the center-of-mass system. 'Legendre' means Gauss-Legendre quadrature, 'Mehler' Gauss-Mehler quadrature (also known as Gauss-Chebyshev quadrature). 'no' means that the scattering angle is determined by interpolation in the scattering table specified by SCAFIL. Either both or none of QUAD and QUADTIME must be 'no'.

Type: index variable array (ATOM2,ATOM1) of character(8)
 Default: 'no'
 Range: 'Legendre', 'Mehler', 'no'

QUADTIME — Method of time integral quadrature

This parameter specifies the method of numerical quadrature for the calculation of the time integral. This parameter is currently unused. Time integrals are calculated using the scattering table SCATTAB.

REFLECTMAX — Maximum number of surface reflections

This parameter specifies the maximum number of reflections at the surface potential. Such reflections may occur when a projectile tries to leave the target but does not have enough energy to do so. When the number of surface reflections of a projectile exceeds REFLECTMAX, the projectile is stopped.

Type: integer
 Default: ∞
 Range: ≥ 0

SCAFIL — Scattering table file

This parameter specifies a file containing tabulated scattering angles and time integrals. This model is used if QUAD='no'. Then the scattering angles and time integrals are determined by bicubic interpolation in the table contained in SCAFIL. Files are assumed to be located in the subdirectory 'scattab' of the search path.

Type: index variable array (ATOM2,ATOM1) of character(80)
 Default: 'BSI', 'SISI', 'PSI', 'ASSI', 'B0', or 'ERSI' for the appropriate atom combinations, 'SCATTAB' otherwise
 Range: in addition to the possible default values, 'SCATTAB_big', or the name of a user-supplied file containing a scattering table

SCRFIL — Screening function file

This parameter specifies a file containing tabulated values of the screening function. This model is used if QUAD specifies a numerical integration method and COEFFIL is either unspecified or overridden by specification of SCRFIL. COEFFIL and SCRFIL must not appear on the same input record. Specification of COEFFIL or SCRFIL on later input records override those on earlier ones. SCRFIL='ZBL' and SCRFIL='DMo1' are interpreted as SCRFIL='zbl/XXYY' and SCRFIL='dmol/XXYY', respectively, where XX and YY are replaced by the chemical element names of the respective atoms. Only atom combinations with the atomic number of XX smaller than or equal to the atomic number of YY are available. Files are assumed to be located in the subdirectory 'screentab' of the search path.

Type: index variable array (ATOM2,ATOM1) of character(80)
 Default: 'ZBL'
 Range: 'ZBL', 'DMo1', 'zbl/XXYY', 'dmol/XXYY' (for the meaning of XXYY see above), or the name of a user-supplied file containing a screening function table

SCRFUN — Screening function

This parameter specifies the screening function for scattering angle calculation. Valid choices are either built-in functions ('ZBL', 'KrC', 'Moliere'), 'coeffil' as a reference to a coefficients file, or 'scrfil' as a reference to a screening function file. In the two latter cases, the file is specified by COEFFIL and SCRFIL, respectively. SCRFUN is ignored if QUAD='no'

Type: index variable array (ATOM2,ATOM1) of character(8)

Default: 'ZBL'

Range: 'ZBL', 'KrC', 'Moliere', 'coeffil', 'scrfil'

SCRLEN — Screening length

This parameter specifies the model for the screening length for the calculation of the scattering angle. The model specifies the dependence on the atomic numbers of the projectile and the target atom. SCRLEN='1' means a screening length of 1. For the calculation of the time integral, always the ZBL screening length is used.

Type: index variable array (ATOM2,ATOM1) of character(8)

Default: '1' for SCRFUN='scrfil'

'ZBL' otherwise

Range: '1', 'ZBL', 'Firsov', 'Lindhard'

5.9 &SEPAR

This record is used to define the parameters of the electronic stopping model. For a description of the parameters refer to Chapter 3.5. The &SEPAR record has the index variables ATOM1 and ATOM2. ATOM1 corresponds to the index of the projectile, while ATOM2 corresponds to the index of the target atom/recoil. Regarding the definition of atom indices, see Section 5.3.

CORLIN — Correction to Lindhard stopping

This parameter specifies the correction factor k/k_L to be applied to the Lindhard stopping formula $S_e = k_L\sqrt{E}$ in the velocity proportional regime. CORLIN is only effective if MODEL=1 or MODEL=3.

Type: index variable array (ATOM2,ATOM1) of real
 Default: see [58] for Si targets
 1.1 for Z1=5 in SiO₂,
 1 otherwise
 Range: ≥ 0

COBETHE — Bethe Bloch stopping power constant

This parameter specifies the Bethe constant c_0 to be applied in calculation of the Bethe-Bloch stopping power. COBETHE is only effective if MODEL=1 or MODEL=3.

Type: index variable array (ATOM2,ATOM1) of real
 Default: 1.43 for Z1=1 [59]
 1 otherwise
 Range: arbitrary

C1BETHE — Bethe Bloch stopping power linear factor

This parameter specifies the linear Bethe factor c_1 to be applied in calculation of the Bethe-Bloch stopping power. C1BETHE is only effective if MODEL=1 or MODEL=3.

Type: index variable array (ATOM2,ATOM1) of real
 Default: $40*Z1*Z2$ for Z1=1 [59]
 $100*Z1*Z2$ for Z1=2
 5 otherwise
 Range: arbitrary

EMINSE — Minimum energy for electronic stopping

This parameter specifies the energy below which electronic stopping vanishes. For energies below **EMINSE** the electronic stopping power S_e is set to zero. For energies above **EMINSE**, **EMINSE** is subtracted from the particle energy for the evaluation of the stopping power. The units of **EMINSE** are eV.

Type: index variable array (**ATOM2,ATOM1**) of real
 Default: 0
 Range: any

ENL — Energies for nonlocal fractions

This parameter specifies the energies E^{nl} to which the nonlocal fraction **XNL** values correspond. The units of **ENL** are eV. **ENL=0** means the energy at the stopping power maximum and is allowed only as the last **ENL** value.

Type: index variable array (**ATOM2,ATOM1**) of character array
 Default: see **XNL**
 Range: ≥ 0 , increasing

ESTRAG — Kink energy for electronic straggling

This parameter specifies the energy where the low- and high-energy asymptotic behaviors of the electronic straggling intersect. At low energies the straggling is proportional to the energy, while at high energies it is constant and equal to the Bohr straggling.

Type: index variable array (**ATOM2,ATOM1**) of real
 Default: 100 keV \times mass of **ATOM1**
 Range: ≥ 0

FACSCR — Factor for the screening length

This parameter specifies the factor f to be applied to the Oen-Robinson screening length if **FAC2SCR** is not specified. If **FAC2SCR** is specified, it specifies f in the low-energy limit.

Type: index variable array (**ATOM2,ATOM1**) of real
 Default: $1.37/Z_1^{1/3}$
 Range: > 0

FAC2SCR — Factor for the screening length at $x^{nl} = 1$

This parameter specifies a linear dependence of the factor f for the screening length on the nonlocal fraction x^{nl} . If FAC2SCR is specified, FACSCR is considered the value of f at $x^{nl} = 0$ and FAC2SCR the value of f at $x^{nl} = 1$.

Type: index variable array (ATOM2,ATOM1) of real
 Default: FACSCR
 Range: ≥ 0

FILSE — Random electronic stopping file

This parameter specifies the name of the random electronic stopping file. FILSE is only effective if MODEL=7.

Type: index variable array (ATOM2,ATOM1) of character(80)
 Default: – (obligatory if MODEL=7)
 Range: any name of a file containing a random electronic stopping.

FIRSOV — Firsov stopping flag

This parameter specifies whether the impact parameter dependent electronic stopping shall be described by the Firsov model. Firsov stopping may be modified by POWFIRS. If FIRSOV='off', the Oen-Robinson model is used instead.

Type: index variable array (ATOM2, ATOM1) of character(3)
 Default: 'off'
 Range: 'on', 'off'

LAPSYS — Apsis flag

This parameter specifies whether the apsis (distance of closest approach) of the collision (LAPSYS=T) or the impact parameter (LAPSYS=F) is used in the local electronic stopping model. Using the apsis is physically more sound, while using the impact parameter may save some computation time.

Type: logical
 Default: T if LRCOIL=T and ELIMKP<100
 F otherwise
 Range: T, F

LSE — Electronic stopping flag

This parameter specifies whether electronic stopping shall be considered.

Type: logical
 Default: T
 Range: T, F

LTAB — Stopping table flag

This parameter specifies whether tables of the nonlocal electronic stopping power shall be written to the file SE for each ion species. The energies are equally spaced on a logarithmic scale between EF of the &SNPAR record and ENERGY of the &IONS record. Units are eV for the energy and eV/Å for the stopping power. For LTAB=T no simulation of ion trajectories is performed.

Type: logical
 Default: F
 Range: T, F

MODEL — Electronic stopping model

This parameter specifies the electronic stopping model. MODEL=1 or 3 denotes the default stopping power model described in Section 3.5. MODEL=2 or 4 indicates that ZBL stopping, MODEL=5 or 6 that KKK stopping should be used instead of Lindhard stopping. MODEL=7 denotes that the electronic stopping will be read from the file FILSE. For MODEL=3, 4, or 6 Eq. 3.19 is used, for MODEL=1, 2, or 5 Eq. 3.18 is used.

Type: index variable array (ATOM2,ATOM1) of integer
 Default: 1 for $E_0 < 10$ keV·ion atom mass,
 3 otherwise
 Range: 1...7

POWFIRS — Power in Firsov model

This parameter specifies the power used in the Firsov model that describes the impact parameter dependence of the electronic stopping. Note that Firsov stopping is only used if FIRSOV='on'.

Type: index variable array (ATOM2,ATOM1) of real
 Default: 5
 Range: > 2

POWINT — Power for Interpolation of Stopping Powers

This parameter specifies the constant used to interpolate the low-energy (Lindhard like) stopping power with the modified Bethe Bloch stopping power. POWINT is only effective if MODEL=1 or 3.

Type: index variable array (ATOM2,ATOM1) of real
 Default: according to [58] for ions specified there
 1 otherwise
 Range: > 0

POWLIN — Power for low energy electronic stopping

This parameter specifies the power p in the energy dependence of the Lindhard stopping power. POWLIN is only effective for MODEL=1 or 3. Note that changing POWLIN will leave the electronic stopping power at 1 eV constant. Since this is probably not what you want, you have to change CORLIN accordingly.

Type: index variable array (ATOM1) of real
 Default: 0.5
 Range: $0 < \text{POWLIN} \leq 1$

PTAB — Impact parameter for stopping power tabulation

This parameter specifies the impact parameter to be used for the local part of the electronic energy loss. It is only used for the stopping power table to be written to the file SE if the flag LTAB is set.

Type: real
 Default: 0
 Range: ≥ 0

STRAGGLE — Electronic straggling flag

This parameter specifies whether electronic straggling according to Konac, Klatt, and Kalbitzer (KKK) shall be taken into account. The KKK model may be modified using the ESTRAG parameter.

Type: index variable array (ATOM2, ATOM1) of character(3)
 Default: 'on' if ESTRAG is specified
 'off' otherwise
 Range: 'on', 'off'

XNL — Nonlocal fraction

This parameter specifies the nonlocal fraction x^{nl} . If **ENL** values are specified, the same number of **XNL** values have to be specified, which then correspond to the respective **ENL** values. If **ENL** values are not specified, not more **XNL** values must be specified than default values of **ENL** exist (usually 2). If only one **XNL** value is given, the nonlocal fraction of electronic stopping is independent of energy.

Type: index variable array (**ATOM2,ATOM1**) of character array

Default: 0.15, 0.32 at 1eV, 1keV for **Z1=5**,
0.25, 0.50 at 1eV, 1MeV for **Z1=15**,
according to [58] for ions specified there,
0.5 otherwise

Range: ≥ 0

5.10 &OUTPUT

This record is used to specify the amount of output to be written to output files.

AMAXA — Analyzer maximum acceptance angle

This parameter specifies the upper limit to the acceptance angle for backscattered or transmitted projectiles. Ions with directions farther away from the analyzer direction than **AMAXA** are not included in the statistics and they are not written to the trajectory file. For the definition of the analyzer direction see **TILTA** and **ROTATA**. The units of **AMAXA** are degrees.

Type: real
Default: 180
Range: $AMINA \leq AMAXA \leq 180$

AMINA — Analyzer minimum acceptance angle

This parameter specifies the lower limit to the acceptance angle for backscattered or transmitted projectiles. Ions with directions closer to the analyzer direction than **AMINA** are not included in the statistics and they are not written to the trajectory file. For the definition of the analyzer direction see **TILTA** and **ROTATA**. The units of **AMINA** are degrees.

Type: real
Default: 0
Range: $0 \leq AMINA \leq AMAXA$

DPOWER — Displacement power in displacement histograms

This parameter specifies the box width variation of displacement histograms (HISD file): Internally, $d^{1/\text{DPOWER}}$ is scored in an equidistant histogram, but the box boundaries are taken to the DPOWER-th power when written to the HISD file, thus the displacements d appear as non-equidistant box boundaries in the HISD file. For DPOWER=1, the histogram is equidistant; for DPOWER>1, box widths are smaller at small energies and larger at large energies. Each histogram box will receive the same number of particles if the displacement distribution is proportional to $d^{-\text{DPOWER}+1}$. Note that the displacement distribution must decay faster than d^{-2} if it is given by a power law. Therefore, for the default value of DPOWER=3 we still expect increasing statistical error with increasing displacement d .

Type: integer
 Default: 3
 Range: > 0

EMAXA — Analyzer maximum acceptance energy

This parameter specifies the upper limit to the acceptance energy, Ions with energies larger than EMAXA are not included in the RBS statistics or in the statistics of transmitted or backscattered ions. The units of EMAXA are keV.

Type: real
 Default: E0
 Range: EMINA ≤ EMAXA

EMINA — Analyzer minimum acceptance energy

This parameter specifies the lower limit to the acceptance energy, i.e., ions with energies lower than EMINA are not included in the RBS statistics or in the statistics of transmitted or backscattered ions. The units of EMINA are keV.

Type: real
 Default: 0
 Range: $0 \leq \text{EMINA} \leq \text{EMAXA}$

EPOWER — Energy power in histogram

This parameter specifies the box width variation of energy histograms for backward and forward sputtering if LHISEB=T and/or LHISET=T: Internally, $E^{1/\text{EPOWER}}$ is scored in an equidistant histogram, but the box boundaries are taken to the EPOWER-th power when written to the HISEB and/or HISET file, thus the displacements E appear as non-equidistant box boundaries in the HISEB and/or HISET file. For EPOWER=1, the histogram is equidistant; for EPOWER>1, box widths are smaller at small energies and larger at large energies. Each histogram box will receive the same number of particles if the energy distribution of the sputtered particles is proportional to $E^{-\text{EPOWER}+1}$. The default value of EPOWER=3 corresponds to the energy distribution predicted by Sigmund.

Type: integer
 Default: 3
 Range: > 0

ITRA — Output indicator for trajectory file

This parameter indicates what output is to be written to the TRA file. Lines with the last value (IFLAG) corresponding to a value of ITRA will be written (see the description of the TRA file in Chapter 4).

Type: character array of integer
 Default: '-1,0,1,2,3,4,5' if LTRA=T, '' otherwise
 Range: $-1 \leq \text{ITRA} \leq 5$

LCELL — Output of dynamic cell target

This parameter specifies whether the contents of the dynamic cell target structure shall be written to the CELL or CELL2 file.

Type: logical
 Default: F
 Range: T, F

LCYL — Output in cylinder coordinates

This parameter specifies whether the 2-D histograms are distributions in cylinder coordinates. The first coordinate is the radius, the second is depth. The axis of the cylinder is parallel to the z axis and has (x,y) coordinates (XINIT,YINIT) from the &IONS record.

Type: logical
Default: **F**
Range: **T, F**

LDIST — Output of distance function

This parameter specifies whether the signed distance from the surface shall be written to the DIST2 file. Only effective if NDIM=2.

Type: logical
Default: **F**
Range: **T, F**

LHIS — Output of 1-D spatial atom distribution

This parameter specifies whether the 1-D histogram of the spatial atom distributions shall be written to the HIS file.

Type: logical
Default: **F**
Range: **T, F**

LHIS2 — Output of 2-D spatial atom distributions

This parameter specifies whether the 2-D histogram of the spatial atom distribution shall be written to the HIS2 file.

Type: logical
Default: **F**
Range: **T, F**

LHIS2B — Output of 2-D backscattered atom spatial distributions

This parameter specifies whether the 2-D histogram of the exit points and points of origin of backscattered atoms shall be written to the HIS2B file.

Type: logical
Default: F
Range: T, F

LHIS2T — Output of 2-D transmitted atom spatial distributions

This parameter specifies whether the 2-D histogram of the exit points and points of origin of transmitted atoms shall be written to the HIS2T file.

Type: logical
Default: F
Range: T, F

LHIS3 — Output of 3-D spatial atom distributions

This parameter specifies whether the histogram of the 3-D spatial atom distribution shall be written to the HIS3 file.

Type: logical
Default: F
Range: T, F

LHISA2B — Output of 2-D backscattered angular distributions

This parameter specifies whether the histogram for the 2-D angular distribution of the backscattered atoms shall be written to the HISA2B file. The independent variables of the histogram are the direction cosines with respect to the x and y axis.

Type: logical
Default: F
Range: T, F

LHISA2T — Output of 2-D transmitted angular distributions

This parameter specifies whether the histogram for the 2-D angular distribution of the transmitted atoms shall be written to the HISA2T file. The independent variables of the histogram are the direction cosines with respect to the x and y axis.

Type: logical
Default: F
Range: T, F

LHISAAB — Output of backscattered angular distributions (3-D)

This parameter specifies whether the histogram for the angular distribution of the backscattered atoms shall be written to the HISAAB file. The angle is measured with respect to the analyzer direction.

Type: logical
Default: F
Range: T, F

LHISAAT — Output of transmitted angular distributions (3-D)

This parameter specifies whether the histogram for the angular distribution of the transmitted atoms shall be written to the HISAAT file. The angle is measured with respect to the analyzer direction.

Type: logical
Default: F
Range: T, F

LHISAB — Output of backscattered angular distributions (2-D)

This parameter specifies whether the histogram for the angular distribution of the backscattered atoms shall be written to the HISAB file. The angle is measured between the projection to the xz plane and the negative z axis.

Type: logical
Default: F
Range: T, F

LHISAT — Output of transmitted angular distributions (2-D)

This parameter specifies whether the histogram for the angular distribution of the transmitted atoms shall be written to the HISAT file. The angle is measured between the projection to the xz plane and the negative z axis.

Type: logical
Default: F
Range: T, F

LHISB — Output of backscattered spatial distributions

This parameter specifies whether the histogram for the spatial distribution of the backscattered atoms shall be written to the HISB file.

Type: logical
Default: F
Range: T, F

LHISD — Output of displacement distributions

This parameter specifies whether the histogram of the atom displacement distribution shall be written to the HISD file. The atom displacements are the distances between the starting and end points of the recoils that do not leave the target.

Type: logical
Default: F
Range: T, F

LHISEB — Output of backscattered energy distributions

This parameter specifies whether the histogram for the energy distribution of the backscattered atoms shall be written to the HISEB file.

Type: logical
Default: F
Range: T, F

LHISEE — Output of electronic energy deposition distributions

This parameter specifies whether the histogram of the 1D electronic energy deposition distribution shall be written to the HISNE file.

Type: logical
Default: F
Range: T, F

LHISEE2 — Output of 2-D electronic energy deposition distributions

This parameter specifies whether the histogram of the 2-D electronic energy deposition distribution shall be written to the HISEE2 file.

Type: logical
Default: F
Range: T, F

LHISEE3 — Output of 3-D electronic energy deposition distributions

This parameter specifies whether the histogram of the 3-D electronic energy deposition distribution shall be written to the HISEE3 file.

Type: logical
Default: F
Range: T, F

LHISET — Output of transmitted energy distributions

This parameter specifies whether the histogram for the energy distribution of the transmitted atoms shall be written to the HISET file.

Type: logical
Default: F
Range: T, F

LHISNE — Output of nuclear energy deposition distributions

This parameter specifies whether the histogram of the 1D nuclear energy deposition distribution shall be written to the HISNE file.

Type: logical
Default: F
Range: T, F

LHISNE2 — Output of 2-D nuclear energy deposition distributions

This parameter specifies whether the histogram of the 2-D nuclear energy deposition distribution shall be written to the HISNE2 file.

Type: logical
Default: F
Range: T, F

LHISNE3 — Output of 3-D nuclear energy deposition distributions

This parameter specifies whether the histogram of the 3-D nuclear energy deposition distribution shall be written to the HISNE3 file.

Type: logical
Default: F
Range: T, F

LHISIV — Output of I–V pair distributions

This parameter specifies whether the histogram for the I–V pair distribution function shall be written to the HISIV file. LHISIV=T requires the capture radius model to be activated (LCAP=T on the *&DAMAGE* record), and is not possible for dynamic simulations (LDYN=T on the *&SETUP* record).

Type: logical
Default: F
Range: T, F

LHISM — Output of first redistributive moment depth distribution

This parameter specifies whether the histogram of the first redistributive moment as a function of depth shall be written to the HISM file. See Section 4.3 for an explanation of the first redistributive moment.

Type: logical
Default: F
Range: T, F

LHISP — Output of the deposited momentum depth distribution

This parameter specifies whether the histogram of the deposited momentum distribution shall be written to the HISP file. See Section 4.3 for an explanation of deposited momentum.

Type: logical
Default: F
Range: T, F

LHIST — Output of transmitted spatial distributions

This parameter specifies whether the histogram for the spatial distribution of the transmitted atoms shall be written to the HIST file.

Type: logical
Default: F
Range: T, F

LMOM — Output of spatial atom distribution moments

This parameter specifies whether the global moments of the ion distribution shall be written to the OUT file. If LDAM=T the moments of the interstitial distribution are written as well. If in addition LRCOIL=T also the moments of the vacancies are written.

Type: logical
Default: F
Range: T, F

LMOMAAB — Output of backscattered atoms angular moments (3-D)

This parameter specifies whether the moments of the angular distribution of backscattered ions and backspattered atoms shall be written to the OUT file. The angle is measured with respect to the analyzer direction.

Type: logical
Default: F
Range: T, F

LMOMAAT — Output of transmitted atoms angular moments (3-D)

This parameter specifies whether the moments of the angular distribution of transmitted ions and forward sputtered atoms shall be written to the OUT file. The angle is measured with respect to the analyzer direction.

Type: logical
Default: F
Range: T, F

LMOMAB — Output of backscattered atoms angular moments (2-D)

This parameter specifies whether the moments of the angular distribution of backscattered ions and backspattered atoms shall be written to the OUT file. The angle is measured between the projection to the xz plane and the negative z axis.

Type: logical
Default: F
Range: T, F

LMOMAT — Output of transmitted atoms angular moments (2-D)

This parameter specifies whether the moments of the angular distribution of transmitted ions and forward sputtered atoms shall be written to the OUT file. The angle is measured between the projection to the xz plane and the negative z axis.

Type: logical
Default: F
Range: T, F

LMOMB — Output of backscattered atoms spatial moments

This parameter specifies whether the moments of the and spatial distribution of backscattered ions and backspattered atoms shall be written to the OUT file.

Type: logical
Default: F
Range: T, F

LMOMEB — Output of backscattered atoms energy moments

This parameter specifies whether the moments of the energy distribution of backscattered ions and backspattered atoms shall be written to the OUT file.

Type: logical
Default: F
Range: T, F

LMOMEE — Output of electronic energy deposition moments

This parameter specifies whether the moments of the electronic energy deposition distribution shall be written to the OUT file.

Type: logical
Default: F
Range: T, F

LMOMET — Output of transmitted atoms energy moments

This parameter specifies whether the moments of the energy distribution of transmitted ions and forward sputtered atoms shall be written to the OUT file.

Type: logical
Default: F
Range: T, F

LMOMNE — Output of nuclear energy deposition moments

This parameter specifies whether the moments of the nuclear energy deposition distribution shall be written to the OUT file.

Type: logical
Default: F
Range: T, F

LMOMT — Output of transmitted ions spatial moments

This parameter specifies whether the moments of the spatial distribution of transmitted ions and forward sputtered atoms shall be written to the OUT file.

Type: logical
Default: F
Range: T, F

LPERX — Periodic boundary condition flag

This parameter specifies whether the x-coordinates of the positions of events that are used for output, are mapped into the periodic boundary interval (XPER of *&geometry* record).

Type: logical
Default: LPERX of *&GEOMETRY* record
Range: T only if LPERX=T of *&GEOMETRY*, F

LRBS — RBS energy spectrum histogram

This parameter specifies whether the RBS simulation is performed and the energy spectrum is written to the RBS file.

Type: logical
Default: F
Range: T, F

LTRA — Output of trajectory

This parameter specifies that information about the trajectories shall be written to the TRA file. If ITRA is specified, only the output corresponding to the values of ITRA are written, and LTRA is ignored. If LTRA=T and ITRA is not specified, all possible output is written. If LTRA=F and ITRA is not specified, no trajectory file is written (regarding the TRA file and its format see Chapter 4).

Type: logical

Default: F

Range: T, F

MCOORD — Coordinate system model

This parameter specifies the model of the coordinate system to be used with the statistics output. The meaning of the model numbers are specified below.

Type: integer

Default: 1

Range: 1... same as in &GEOM
2... aligned with the ion's incidence direction
3... aligned with direction given by TILT and ROTATE
4... aligned with direction given by TILTA and ROTATA
5... crystal coordinate system

NBOX — Maximum number of 1-D histogram boxes

This parameter specifies the maximum number of boxes in 1-D histograms. If zero, there is no upper bound to the number of boxes, and the box width is kept equal to WBOX.

Type: integer

Default: 32

Range: ≥ 0

NBOX2 — Maximum number of 2-D histogram boxes

This parameter specifies the maximum number of boxes in 2-D histograms. The first value corresponds to the x-direction, the second value corresponds to the z-direction. If zero, there is no upper bound to the number of boxes in that direction, and the box width is kept equal to `WBOX2`. Either all or none of `NBOX2` must be zero.

Type: simple array (2) of integer
Default: 36, 24 for point responses
Range: ≥ 0

NBOX3 — Maximum number of 3-D histogram boxes

This parameter specifies the maximum number of boxes in 3-D histograms. The first value corresponds to the x-direction, the second to the y-direction, and the third one to the z-direction. If zero, there is no upper bound to the number of boxes, and the box width is kept equal to `WBOX3`. Either all or none of `NBOX3` must be zero.

Type: simple array (3) of integer
Default: 36, 36, 24
Range: ≥ 0

NBOXA — Maximum number of angular histogram boxes

This parameter specifies the maximum number of boxes in angular histograms (with respect to the analyzer direction) in the backscattered/transmitted ions statistics. The maximum number of boxes in angular histograms of the 2-D projection will be $2*NBOXA$. If zero, there is no upper bound to the number of boxes, and the box width is kept equal to `WBOXA`.

Type: integer
Default: 90
Range: ≥ 0

NBOXA2 — Maximum number of 2-D angular histogram boxes

This parameter specifies the maximum number of boxes in 2-D angular histograms in the backscattered/transmitted ions statistics. If zero, there is no upper bound to the number of boxes, and the box width is kept equal to `WBOXA2`.

Type: simple array (2) of integer
Default: 50, `NBOXA2(1)`
Range: ≥ 0

NBOXD — Maximum number of displacement and pair histogram boxes

This parameter specifies the maximum number of boxes in displacement and pair histograms (HISD, HISIV, and HISIV2 files). If zero, there is no upper bound to the number of boxes, and the box width is kept equal to WBOXD.

Type: integer
Default: 100
Range: ≥ 0

NBOXE — Maximum number of energy histogram boxes

This parameter specifies the maximum number of boxes in energy histograms in the backscattered/transmitted ions statistics. If zero, there is no upper bound to the number of boxes, and the box width is kept equal to WBOXE.

Type: integer
Default: 100
Range: ≥ 0

NHISIV — Number of pair distribution histograms

This parameter specifies the number of pair distribution histograms. The positions of point defects considered may be restricted for each histogram to a range in depth by POSIVMIN and POSIVMAX.

Type: integer
Default: required if LHSIV=T or LHSIV2=T
Range: $1 \leq \text{NHISIV} \leq 10$

NIONBK — Number of ions for backup

This parameter specifies that a backup file shall be written/updated after every NIONBK ions. NIONBK=0 means that no backup file shall be written. In case NIONBK>NION, a backup file is written after the last ion. NIONBK cannot be specified in case of a dynamic simulation (LDYN=T).

Type: integer
Default: 0
Range: ≥ 0

NIONTRA — Number of ions for trajectory output

This parameter specifies that a trajectory shall be written to the TRA file for every NIONTRAth ion. NIONTRA=1 means that trajectory output shall be written for every ion. In case NIONTRA>NION, a backup file is written after the last ion. This parameter can be used for debugging to limit the trajectory output to the cascade of interest.

Type: integer
Default: 1
Range: > 0

NLATMOM — Maximum order of lateral moments

This parameter specifies the maximum order of the calculated lateral moments. NLATMOM=0 means there is no output of lateral moments.

Type: integer
Default: 2
Range: $0 \leq \text{NLATMOM} \leq 4$

NMIXMOM — Maximum order of mixed moments

This parameter specifies the maximum order of the calculated mixed moments. NMIXMOM=0 or NMIXMOM=1 means there is no output of mixed moments.

Type: integer
Default: 0
Range: $0 \leq \text{NMIXMOM} \leq 4$

POSIVMAX — Maximum depth for pair distribution histogram

This parameter specifies the maximum depth of the point defects to be considered for each pair distribution histogram.

Type: simple array (NHISIV) of real
Default: ∞
Range: POSIVMIN < POSIVMAX

POSIVMIN — Minimum depth for pair distribution histogram

This parameter specifies the minimum depth of the point defects to be considered for each pair distribution histogram.

Type: simple array (NHISIV) of real
 Default: 0
 Range: POSIVMIN < POSIVMAX

ROTATA — Analyzer rotation angle

This parameter specifies the rotation angle of the sample with respect to the analyzer used for transmission and backscattering statistics. For the meaning of the analyzer direction see TILTA. ROTATA is defined analogously to ROTATE. The units of ROTATA are degrees.

Type: real
 Default: 0

SHORTTRA — Shorten trajectory file

This parameter is a flag, if set, specifying that collision points along the trajectory (IFLAG=2) are only written to the TRA file if a recoil starts motion in this collision. SHORTTRA is ignored unless ITRA='..,2,..' or LTRA=T is specified.

Type: logical
 Default: F
 Range: T, F

TILTA — Analyzer tilt angle

This parameter specifies the tilt angle of the sample with respect to the analyzer direction. The analyzer direction is the reference direction for AMINA and AMAXA. In addition, it defines the statistics coordinate system if MCOORD=4. For RBS statistics, it is used independent of the value of MCOORD. For backscattering statistics and RBS, the analyzer is just opposite the direction specified by TILTA and ROTATA, i.e., in these cases TILTA=0 means the analyzer is in negative z direction, while otherwise it is in positive z direction. In particular, for RBS, TILTA=0 means the RBS detector is located along the negative z axis (180° backscattering angle). The units of TILTA are degrees.

Type: real
 Default: 0
 Range: $0 \leq \text{TILTA} \leq 90$

WBOX — Initial 1-D histogram box width

This parameter specifies the initial box width of all 1-D histograms. If `NBOX=0`, the box widths remain constant throughout the simulation. Otherwise, they are doubled whenever necessary to accommodate all scores in `NBOX` boxes. The units are Å. The value of `WBOX` might be slightly adjusted by `IMSIL` if `LVAC=T` is set.

Type: real
Default: 100/128
Range: > 0

WBOX2 — Initial 2-D histogram box widths

This parameter specifies the initial box widths of all 2-D histograms. The first value corresponds to the box width in x direction, the second value corresponds to the box width in z direction. If `NBOX2=0`, the box widths remain constant throughout the simulation. Otherwise, they are doubled whenever necessary to accommodate all scores in `NBOX2` boxes. The values of `WBOX2` might be slightly adjusted by `IMSIL` if `LVAC=T` is set. The units are Å.

Type: simple array (2) of real
Default: `WBOX`, `WBOX2(1)`
Range: > 0

WBOX3 — Initial 3-D histogram box widths

This parameter specifies the initial box widths of the 3-D histograms. The first value corresponds to the box width in x direction, the second to the box width in y direction, and the third one to the box width in z direction. If `NBOX3=0`, the box widths remain constant throughout the simulation. Otherwise, they are doubled whenever necessary to accommodate all scores in `NBOX3` boxes. The values of `WBOX3` might be slightly adjusted by `IMSIL` if `LVAC=T` is set. The units are Å.

Type: simple array (3) of real
Default: `WBOX`, `WBOX3(1)`, `WBOX3(1)`
Range: > 0

WBOXA — Initial angular histogram box width

This parameter specifies the initial box width of all angular histograms for the backscattered/transmitted ions statistics. If NBOXA=0, the box widths remain constant throughout the simulation. Otherwise, they are doubled whenever necessary to accomodate all scores in NBOXA boxes. The units are degrees.

Type: real
 Default: 1
 Range: > 0

WBOXA2 — Initial 2-D angular histogram box width

This parameter specifies the initial box widths of all 2-D angular histograms for the backscattered/transmitted ions statistics. If NBOXA2=0, the box widths remain constant throughout the simulation. Otherwise, they are doubled whenever necessary to accommodate all scores in NBOXA2 boxes. Note that the independent variables of 2-D angular distributions are the direction cosines with respect to the x and y axis.

Type: simple array (2) of real
 Default: 0.01, WBOXA2(1)
 Range: > 0

WBOXD — Initial displacement histogram box width

This parameter specifies the initial box width of the displacement histograms (HISD file). If NBOXD=0, the box widths remain constant throughout the simulation. Otherwise, they are doubled whenever necessary to accomodate all scores in NBOXD boxes. The units are Å if DPOWER=1. If DPOWER>1, the box width applies to $d^{1/DPOWER}$, and the units of WBOXD are Å^{1/DPOWER}. As a result, the right histogram edges are at $d = (i \cdot \text{WBOXD})^{DPOWER}$ where i are integers.

Type: real
 Default: 0.1
 Range: > 0

WBOXE — Initial energy histogram box width

This parameter specifies the initial box width of all energy histograms for the RBS/backscattered/transmitted ions statistics. If NBOXE=0, the box widths remain constant throughout the simulation. Otherwise, they are doubled whenever necessary to accomodate all scores in NBOXE boxes. The units are eV if EPOWER=1. If EPOWER>1, the box width applies to $E^{1/EPOWER}$, and the units of WBOXE are $\text{eV}^{1/EPOWER}$. As a result, the right histogram edges are at $E = (i \cdot \text{WBOXE})^{EPOWER}$ where i are integers.

Type: real
Default: ENERGY**(1/EPOWER)/NBOXE
Range: > 0

Chapter 6

Auxiliary Scripts

IMSIL comes with a number of Python scripts that facilitate running the program and reading and plotting the output data. They are found in the `scripts` directory. The scripts require Python 3.

6.1 Scripts for Running IMSIL

There are two scripts for running IMSIL: `imsil.py` for running a single simulation from one input file, and `mimsil.py` for running multiple simulations whose input files are generated automatically from a template and a parameters file. It is also possible to pass several input files to `imsil.py` or several pairs of template and parameters files to `mimsil.py`.

`imsil.py` takes a file with extension `.inp` (the input file), copies it to a directory with the same name but without the extension, renames it to `INP`, runs it there and copies any output files back to the directory where the original version of the input file resides. The output files are again renamed so their names are converted to lowercase and become their file extension. E.g., if `imsil.py` is run with the input file `example.inp`, a directory `example` is created, the input file is copied to `example/INP`, IMSIL is run in the directory `example`, generates `example/OUT`, and, e.g., `example/HIS`. These files are then copied to `example.out` and `example.his`, and the directory `example` is deleted.

A typical call of IMSIL from the directory where the input file `example.inp` resides, reads

```
<path-to-imsil.py> -b example.inp
```

This uses the shebang (first) line of the `imsil.py` script to call the Python 3 interpreter (otherwise you have to call the Python interpreter with the above line as arguments). The extension `.inp` in the specification of the input file may be omitted. `-b` is one of several possible flags and causes IMSIL to run in the background, so you can continue to use your command window.

You can also start several simulations at once:

```
<path-to-imsil.py> -b -p 2 example1.inp example2.inp
```

`-p 2` causes the two simulations to run in parallel. Without the `-p` option they will run one after the other. More generally, `-p` defines the maximum number of simulations that run in parallel. If there are more input files than specified by the `-p` option, the remaining simulations have to wait until a previous simulation terminates.

If you have a passwordless ssh connection to a remote host, you can also delegate the simulation to the remote host:

```
<path-to-imsil.py> -b -r <remote-hostname> example.inp
```

Replace `<remote-hostname>` with the IP address or hostname of the remote host. For this to work, the remote host must be reachable from the local host with `ssh <remote-hostname>`, and from the remote host the local host must be reachable with `ssh <local-hostname>`. The ssh connection must be passwordless in both directions. In addition, the shebang line of `imsil.py` must be valid on the remote host, i.e., the Python 3 interpreter must be called on the remote host with the same command as on the local host. During the operation, the IMSIL executable is copied to the remote host. IMSIL therefore need not be installed on the remote host, but the executable must be compatible with the remote host's processor.

A file with the same name as the input file but extension `.post` can be provided. It must contain valid Python code. If such a file exists, `imsil.py` runs the code after the IMSIL simulation has completed. It can be used to extract parameters from the output data such as the sputter yield.

For a full list of options to the `imsil.py` script see its docstring.

For `mimsil.py`, a template file with extension `.tem` and a parameters file with extension `.par` must be provided. The template file is an input file where the parameters to be varied between the simulations are replaced by variables. Variables are defined by hash signs prepended and appended to their names. Usually, the names are taken identical to the parameter names. E.g., to vary the tilt angle, one writes `tilt=#tilt#` on the `&ions` record. The tilt angle may then be referenced by `tilt` in the parameters file.

The parameters file contains the definition of the parameters and of the input file name for each simulation. There are two formats of the parameters file. In the first format the parameters are defined as a list under each file name:

```
FILE: <filename1>
<parname11>=<value11>
<parname12>=<value12>
...
FILE: <filename2>
<parname21>=<value21>
<parname22>=<value22>
```

```
...
...
```

Here, `filename<i>` denotes the filename of the input and output files without extension, `parname<ij>` the variable names as defined in the template file, and `value<ij>` the values to be substituted for the variables. As an example, let's assume that two simulations are to be performed, one for a tilt angle of 0° and one for a tilt angle of 10° . The template and parameters file are `example.tem` and `example.par`, respectively. The parameters file may read

```
FILE: example_0
tilt=0
FILE: example_10
tilt=10
```

`mimsil.py` produces the input files `example_0.inp` and `example_10.inp`, where the variable `#tilt#` of the template file is replaced by 0 and 10, respectively, and runs the two simulations.

The alternative format of the parameters file defines a Python function `get_params` whose single argument is a dictionary which defines the filenames and parameters. In the above example, the parameters file could alternatively read

```
def get_params(param):
    for tilt in (0, 10):
        fname = 'example_' + str(tilt)
        param[fname] = {}
        param[fname]['tilt'] = str(tilt)
```

On entry, `param` is an empty ordered dictionary. Within the function, the elements of the dictionary must be defined as dictionaries with the keys equal to the parameter names and values equal to the parameter values, converted to strings. The outer dictionary has the file names as keys. Note that the `def` statement must be the first in the file, otherwise the format will not be correctly identified by `mimsil.py`. Therefore, e.g., imports such as `import math` must be made inside the function. This second format of the parameters file is useful when the number of simulations to be defined is great.

In both cases, `mimsil.py` is run with `example.tem` and `example.par` as template and parameters file by

```
<path-to-mimsil.py> -b example.tem
```

Similar as before, `<path-to-mimsil.py>` denotes the path to the `mimsil.py` script. The extension `.tem` may be omitted or replaced by `.par`.

The `mimsil.py` script imports `imsil.py` as a module and passes its options to the main function of `imsil.py`. The `-p` and `-r` options may therefore be used just as with `imsil.py`.

A template file with extension `.ptem` can also be provided for postprocessing. The `.ptem` file contains Python code to be run after each simulation. The same variables as in the

`.tem` file (in the above example `#tilt#`) can be defined and will be replaced by the values defined in the `.par` file. The resulting code will be saved in a `.post` file for each input (`.inp`) file.

6.2 Scripts for Reading and Plotting Results

The `read_output.py` file may be used as a module to access the IMSIL output data from a Python 3 script. In addition, `read_output.py` may be used as a script with simple plotting functionality.

To use functions of `read_output.py` in your own Python script, you must import `read_output` as a module. E.g., to read and plot data from `example.his`, you can write

```
import sys
import matplotlib.pyplot as plt
sys.path.insert(0, <path-to-scripts>)
from read_output import read_his
x, c = read_his('example.his')
plt.plot(x, c)
plt.show()
```

Here, `<path-to-scripts>` denotes the path to the directory where `read_output.py` resides.

The following functions are defined in `read_output.py`:

Function	Purpose
<code>read_par</code>	Read the file names (without extension) defined in a parameters file. Optionally, also return the value of one parameter for each file name.
<code>read_inp</code>	Read the value of a parameter from an input (<code>.inp</code>) file.
<code>read_out</code>	Read a value from an output data table in the output (<code>.out</code>) file.
<code>extract_yield</code>	Read the sputter yield from the output (<code>.out</code>) file.
<code>read_his</code>	Read a 1-D histogram file (<code>.his</code> and other files described in Section 4.3).
<code>read_his2</code>	Read a 2-D histogram file (<code>.his2</code> and other files described in Section 4.4).
<code>read_his3</code>	Read a 3-D histogram file (<code>.his3</code> and other files described in Section 4.5).
<code>read_trajectories</code>	Read a trajectory <code>.tra.</code> file. The return value is a list of cascades, where each cascade is a list of trajectories, where each trajectory is a list of collision points.

For a detailed description of the function arguments and return values see the code.

`read_output.py` may also be called as a script such as

```
<path-to-read_output.py> example.his
```

This has very limited functionality, though. The code in the main function of `read_output.py` is mainly intended to provide a template for calling the functions in your own script.

Chapter 7

Examples

7.1 Ion Implantation into a Flat Target

The following example shows the INP file for the simulation of an arsenic implantation into (100) silicon through a 10 Å native oxide. All recoils are followed.

```
IMSIL test example
Point defects after 10 keV As implantation
&setup    nr=2 natom=3 /
&ions     name='As' energy=10 dose=1e13 tilt=7 rotate=15 diverg=0.3
          nion=10000 /
&material region=1 name='SiO2' xtal='no' /
&material region=2 name='Si' xtal='yes' /
&damage   ldam=t lrcoil=t /
&geometry posif='-10,0,100000' /
&output   lmom=t lhis=t /
```

As mentioned in Chapter 5, the first two lines appear outside any input record and therefore have no further meaning. They serve as a comment. The third line

```
&setup    nr=2 natom=3 /
```

specifies that there are 2 regions and a total of 3 atom species (As ion plus Si and O atoms in the target, see below). Since `ndim` is not specified, its default value `ndim=1` is taken, and a 1-D geometry is assumed (see also the `&geometry` record). The next line

```
&ions     name='As' energy=10 dose=1e13 tilt=7 rotate=15 diverg=0.3
          nion=10000 /
```

defines the properties of the ions: The ion species is As (notice the quotes for character input), the energy 10 keV, the dose 10^{13} cm⁻², the tilt and rotation angles 7° and 15°, respectively, and the beam divergence 0.3°. Since the divergence model is not specified,

its default is used, i.e., isotropic within a maximum angle of `diverg`. 10000 projectiles are requested to be simulated. The next records

```
&material region=1 name='SiO2' xtal='no' /
&material region=2 name='Si' xtal='yes' /
```

specify the materials of the target regions, amorphous SiO₂ (region 1) and crystalline Si (region 2). Since the input file contains no `&crystal` record, the default crystal structure (diamond) and orientation (100) are assumed.

The next record,

```
&damage ldam=t lrcoil=t /
```

specifies that damage in the crystalline region shall be taken into account dynamically for the simulation of subsequent trajectories, that all recoils shall be followed, and that histograms for interstitials and vacancies shall be recorded.

Since there are no `&snpar` and `&separ` records, the default parameters for the nuclear and electronic stopping are used. The next input record

```
&geometry posif='-10,0,100000' /
```

specifies the positions of the interfaces between the target layers (a 1-D geometry is already specified by `ndim` of the `&setup` record). The first layer (SiO₂ as specified in the `&material` record) extends from $z = -10 \text{ \AA}$ to $z = 0$ and the second (Si) from $z = 0$ to $z = 100000 \text{ \AA}$ ($= 10 \mu\text{m}$). Finally,

```
&output lmom=t lhis=t /
```

specifies the amount of output: Dopant and damage moments into the OUT file and 1-D histograms of the dopants and the damage into the HIS file.

If the script delivered with the program is used, the input file could be named `as10.inp`. IMSIL would then produce the files `as10.out` and `as10.his`.

7.2 Ion Implantation into a Trench

This example shows the template and parameters file for the simulation of boron implantations into trenches with uneven surfaces. It shows two new features: The definition of a 2-D geometry, and the use of a template and parameters file to automatically generate two input files.

The template file reads:

```
Boron implantation into a bumpy 2D silicon trench
B in #text#-Si, 25 keV, 7 tilt
&setup      ndim=2 natom=2 nr=1 /
&ions       name='B' energy=25 dose=1e15 tilt=7 diverg=0
```

```

        nion=100000 xinit=-10000,10000 /
&material region=1 name='Si' xtal=#xtal# /
&geometry point=1 pos=-15000,0 /
&geometry point=2 pos=-15000,30000 /
&geometry point=3 pos=15000,30000 /
&geometry point=4 pos=15000,670 /
&geometry point=5 pos=5830,670 /
&geometry point=6 pos=5000,3500 /
&geometry point=7 pos=4830,5000 /
&geometry point=8 pos=5000,8330 /
&geometry point=9 pos=4830,10580 /
&geometry point=10 pos=4420,14000 /
&geometry point=11 pos=4250,17170 /
&geometry point=12 pos=4170,21000 /
&geometry point=13 pos=4000,22830 /
&geometry point=14 pos=2670,23500 /
&geometry point=15 pos= 830,23580 /
&geometry point=16 pos=-1830,23500 /
&geometry point=17 pos=-3500,23170 /
&geometry point=18 pos=-4330,22750 /
&geometry point=19 pos=-4500,20830 /
&geometry point=20 pos=-4580,17580 /
&geometry point=21 pos=-4500,13830 /
&geometry point=22 pos=-4580,9170 /
&geometry point=23 pos=-4170,3750 /
&geometry point=24 pos=-4170,2420 /
&geometry point=25 pos=-4580,0 /
&geometry region=0
  points='1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25' /
&geometry region=1
  points='1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25' /
&output  lmom=t lhis=t nbox2=0,0 wbox2=500,500
        lhis2=t nionbk=10000 / nbox2=100,100

```

The parameter `ndim=2` on the `&setup` record specifies that a 2-D geometry will be defined. On the `&ions` record, `xinit=-10000,10000` specifies that the reference points for the initial ion position are uniformly distributed between $x = -10000 \text{ \AA}$ and $x = 10000 \text{ \AA}$. Since `yinit` and `zinit` are not specified, their default values 0 are assumed.

The `&geometry` records with `point` index variables define the vertices of the polygons describing the region boundaries. E.g., `pos=-15000,0` specifies the vertex ($x = -15000 \text{ \AA}$, $z = 0$). The `&geometry` records with `region` index variables define the region boundaries of the target by connecting the points with the given indices in the given order. `region=0` denotes the circumference of the whole simulation domain. Since we have only one region in this simulation, it is identical to the polygon defining the

region.

Note that we have two variables in this example: `text` and `xtal` indicated by the `#` signs. Their values are specified in the parameters file:

```
FILE: btrench1
text=a
xtal='no'
```

```
FILE: btrench2
text=<100>
xtal='yes'
```

The input files that will be generated by `mimsil.py`, will be `btrench1.inp` and `btrench2.inp`.

7.3 Ion Implantation into a Multilayer Structure

This example shows a trick to realize a target with two planar crystalline layers by connecting the two layers outside the implantation area in a 2-D setup.

The input file reads:

```
B implantation in a multilayer structure using the "2D trick"
B 30 keV, 7 tilt, a/c/a/c
&setup      ndim=2 natom=3 nr=3 /
&ions       name='B' energy=30 dose=5.E13 tilt=7. rotate=15.
            diverg=0.3 nion=50000 /
&material   region=1 name='SiO2' xtal='no' /
&material   region=2 name='Si' xtal='yes' wafer=0.,0.,1. vsurf=1.,1.,0. /
&material   region=3 name='SiO2' xtal='no' /
&geometry   point=1 pos=-3000,0 /
&geometry   point=2 pos=-3000,80 /
&geometry   point=3 pos=-3000,3000 /
&geometry   point=4 pos=3000,3000 /
&geometry   point=5 pos=3000,480 /
&geometry   point=6 pos=-2900,480 /
&geometry   point=7 pos=-2900,380 /
&geometry   point=8 pos=3000,380 /
&geometry   point=9 pos=3000,80 /
&geometry   point=10 pos=3000,0 /
&geometry   region=0 points='1,2,3,4,5,8,9,10' /
&geometry   region=1 points='1,2,9,10' /
&geometry   region=2 points='2,3,4,5,6,7,8,9' /
```

```
&geometry region=3 points='7,6,5,8' /
&output  lmom=t lhis=t lhis2=t /
```

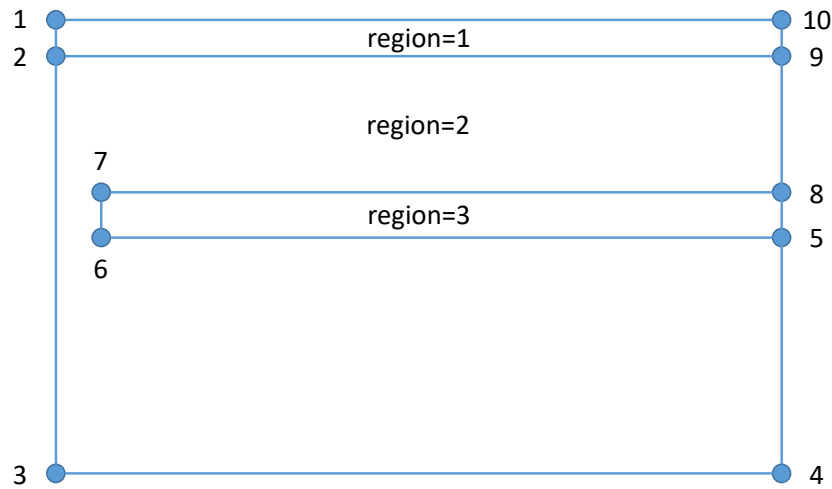


Figure 7.1: Geometry of the multilayer example. Lengths are not to scale. The point and region indices are indicated.

Here, three regions are defined (`nr=3` on the `&setup` record, three `&material` and four `&geometry` records). The geometry defined by the input file is visualized in Fig. 7.1. Since `xinit` is not specified on the `&ions` record, its default value 0 is used. That means, all ions are implanted in the center of the target at $x = 0$, and their lateral straggle will not suffice for them to reach beyond points 6 and 7 laterally.

In the definition of the material of region 2, the orientation of the crystal with respect to the x and z axes is specified: The z axis is parallel to the [001] direction (`wafer=0., 0., 1.`), and the x axis is parallel to [110] (`vsurf=1., 1., 0.`).

7.4 Spot Burn

This example shows the simulation of an ideally focused ion beam on a crystalline Si target covered by a thin SiO₂ layer.

The input file reads:

```
B spot burn in 100-Si, 2 keV, 7 tilt, dose=1
&setup  natom=3 nr=2 /
&ions   name='B' energy=2 dose=1 doseunits='1'
        tilt=7 diverg=0.3 nion=1000000 /
&material region=1 name='SiO2' xtal='no' /
&material region=2 name='Si' xtal='yes' wafer=0,0,1 vsurf=1,0,0 /
&snpar  lstffp=f /
```

```

&damage   ldam=t lrcoil=t freq=0.3 ndim=3 /
&geometry posif='0,10,10000' /
&output   lmom=t lhis=t lhis2=t lhis3=t
          nbox=200 wbox=10
          nbox2=100,50 wbox2=20,20
          nbox3=0,0,0 wbox3=20,20,20 /

```

The dose is specified as one ion (`dose=1 doseunits='1'` on the `&setup` record). This means that the 3-D histograms will represent the probability density functions for the stop position of the ions, and for the positions of the vacancies and interstitials. Since `xinit`, `yinit`, and `zinit` are not specified, these parameters assume their default value 0, so the reference point will be the origin of the coordinate system for all ions.

Note that in this example the x axis is parallel to the $[100]$ crystal axis (`vsurf=1,0,0` on the second `&material` record). `ndim=3` on the `&damage` record means that for dynamic damage consideration the 3-D damage distribution will be taken into account (“damage consideration” means that atoms in the crystalline region are randomly displaced, which normally results in a reduction of the channeling effect). This corresponds to the actual situation in a spot burn. If `ndim=1` were specified, a 1-D damage distribution would be taken into account derived from the implant simulation. Damage-wise this would correspond to an implantation with a homogeneous beam. The 3-D histograms would then correspond to the probability density functions for one ion in a homogeneous beam.

On the `&output` record, output of 1-D, 2-D, and 3-D histograms are requested. Only the 3-D histogram corresponds to the actual physical situation, since the 3-D damage distribution is taken into account. The 1-D and 2-D histograms are just convolutions of the 3-D histogram and do not correspond to line or area implantations.

On the `&output` record, also the two possibilities for specifying the box sizes of the histograms are demonstrated. For the 1-D and 2-D histograms, `nbox` and `nbox2` are nonzero. This means that the number of histogram boxes will never exceed the specified values. The box widths are initialized with the values given by the `wbox` and `wbox` parameter, respectively. The histogram is moved and/or the box widths are doubled so that all entries are captured. In contrast, `nbox3=0`, which means the histogram will always have the box widths specified by `wbox3`, and boxes will be added as to cover all entries.

7.5 Static Sputtering

This example demonstrates a static sputtering simulation of Si by Ar ions. “Static” means that the target is not modified during the simulation.

The input file reads:

```

Sputtering of a-Si, static mode
Ar on a-Si, 10keV, 60 tilt

```

```

&setup      ndim=1 natom=2 nr=1 ldyn=f lsput=t /
&ions       name='Ar' energy=10 dose=1 tilt=60 diverg=0
            nion=100000 /
&material   region=1 name='Si' xtal='no' /
&snpar      chkffpcyl=t /
&damage     lrcoil=t esurf=4.1 ed=15 /
&geometry   posif='0,1000' /
&output     lhis=t wbox=20 nbox=200 lmomab=t lhisab=t wboxa=1 /

```

On the `&setup` record, `lsput=t` is specified, indicating that accurate sputtering simulation is requested. On the `&damage` record, the surface binding energy and the displacement energy are specified. Since accurate sputtering simulation is requested, free flight paths will be chosen such that no recoils with energy larger than the surface binding energy are disregarded near the surface. Deeper in the bulk, larger free flight paths are chosen, as recoils with energy up to the displacement energy can be neglected.

The sputter yield is written to the OUT file. In addition, the angular distribution of the backward sputtered atoms is written to the HISAB file (`lhisab=t`). The box width of the histogram is 1° (`wboxa=1`).

7.6 Dynamic Sputtering

This example demonstrates a dynamic sputtering simulation of Si by Kr ions. In this simulation, the target is modified as the ions bombard the target. To store the state of the target at intermediate times, simulations are run repeatedly, each simulation reading in the output of the previous one. When starting the simulations with `mimsil.py`, the `-p` option must specify the number of parallel processes to equal 1 (which is the default) in order that the simulations run in sequence. Note that the order of the simulations as specified in the parameters file is followed for `-p 1`.

The template file reads:

```

2 keV Kr in Si, tilt=60
&setup      ndim=1 natom=2 nr=1 ldyn=t lsput=t
            usecell=#usecell# filcell='#filcell#' /
&ions       name='Kr' energy=2 dose=1e15 tilt=60 diverg=0 nion=#nion# /
&material   region=1 name='Si' xtal='no' /
&snpar      ef=4.7 denffp=4.7 /
&damage     lrcoil=t ldam=f ed=15. frec=1 /
&damage     atom1=1 esurf=0.1 /
&separ      xnl=0. corlin=1. facscr=1. /
&geometry   posif='0,200' wrough=2 nionupd=100 posfix=190 fracupd=0. /
&output     lhis=t lmomne=t lmomb=t lmomt=f lcell=t wbox=1 nbox=205 /

```

Here we demand dynamic simulation by setting `ldyn=t` on the `&setup` record. As in the static sputtering simulation, `lsput=t` is set to invoke accurate models for sputtering.

In order to read the results of the previous simulation, the `usecell` and `filcell` parameters are used on the `&setup` record. The value of `usecell` must be a variable, since in the first simulation there is no cell file to be read, so `usecell=f` has to be used, while `usecell=t` must be specified for the subsequent simulations. The value of `filcell` must be a variable, too, since the name of the cell file to be read changes from simulation to simulation.

In order that the cell file is written, `lcell=t` is specified on the `&output` record.

There are two `&damage` records. The first applies to all atoms. It is important to specify `lrcoil=t` here in order to allow a sputtering simulation. The second `&damage` record applies only to atom 1, i.e., the first (and only) ion atom, Ar. Its surface binding energy is set to 0.1 eV. Such a low value leads to slow simulations, since all recoils down to 0.1 eV must be simulated; very short free flight paths and large impact parameters must therefore be used. The motivation of this setting is that Ar atoms close to the surface may leave the target easily.

On the `&geometry` record, `wrough=2` specifies a 2 Å thick roughness layer. `nionupd=100` specifies that after every 100 simulated ions relaxation of the target densities is attempted (an actual update is made only if the relative change in any cell density exceeds `FRACUPD`).

```
def get_params(param):
    fname = 'krdyn_1e15'
    param[fname] = {}
    param[fname]['usecell'] = 'f'
    param[fname]['filcell'] = ''
    param[fname]['nion'] = '1000'

    for dose in range(2,10):
        fname = 'krdyn_' + str(dose) + 'e15'
        param[fname] = {}
        param[fname]['usecell'] = 't'
        param[fname]['filcell'] = '../krdyn_' + str(dose-1) + 'e15.cell'
        param[fname]['nion'] = '1000'

    fname = 'krdyn_10e15'
    param[fname] = {}
    param[fname]['usecell'] = 't'
    param[fname]['filcell'] = '../krdyn_9e15.cell'
    param[fname]['nion'] = '10000'
```

The parameters file is written as a Python function in order to allow a Python loop to implement the loop over the simulations. The first simulation must be kept separate as there is no cell file to be read. The last simulation is also kept separate in order to obtain

better statistics for the final results (assuming sputter equilibrium has been reached). Note that `fname` defines the file names of the input and output files (without extension). It is made different for each simulation by the inclusion of the cumulated ion dose.

7.7 RBS

This example shows the input file for the simulation of an RBS/C spectrum of a 1 μm thick (100) crystalline silicon membrane with a damage profile defined in a histogram input file. 3 MeV He ions aligned with the [001] direction are used.

RBS/C spectrum calculation

```
He in <100> Si, 3000 keV, 0 Tilt, 3D interstitial
&setup      nr=1 natom=2
             atom1=2 atom2=2 ndim=1 usehis=t filhis='../dam2percent.his' /
&ions       name='He' energy=3000 dose=1e14 tilt=0.0 nion=10000 /
&material   region=1 name='Si' xtal='on' /
&damage     ldam=t lamo2=f /
&geometry   posif='0,10000' /
&output     lhis=t lrbs=t wboxe=6000. nboxe=0 tilta=10. /
```

The histogram file is read on the `&setup` record (`usehis=t filhis='../dam2percent.his'`). The `.his` file must conform to the format of a 1-D histogram file (see Section 4.3), and can of course be an output histogram of another IMSIL run. In this case it has been created by hand with a constant damage concentration of 2% (not shown). Note that `filhis` must refer to the parent directory if IMSIL is run with the `imsil.py` script, since this file is not copied into the subdirectory where IMSIL is run. `atom1=2` indicates that the second atom stored in the histogram file is to be read. `atom2=2` indicates that the damage concentrations of atom 2 of the simulation are to be initialized with these values.

`lamo2=f` on the `&damage` record specifies that the 3-D random interstitial model be used. In this model, the target atoms are displaced due to lattice vibration according to a 3-D Gaussian probability density function. The default is displacement in the plane perpendicular to the direction of motion according to a 2-D Gaussian probability density function (`lamo2=f`), which is computationally slightly more efficient.

Finally, `lrbs=t` on the `&output` record causes an RBS file to be written, see Section 4.3. The energy box widths are 6000Å. The backscattering angle is specified as 170° (`tilta=10.`).

Bibliography

- [1] G. Hobler. Monte Carlo simulation of two-dimensional implanted dopant distributions at mask edges. *Nucl. Instrum. Meth. B*, 96:155–162, 1995.
- [2] G. Hobler and H. Pötzl. The effect of a screening oxide on ion implantation studied by Monte Carlo simulations. *COMPEL*, 11(4):403–411, 1992.
- [3] G. Hobler, J. Bevk, and A. Agarwal. Channeling of low-energy implanted ions through the poly-Si gate. *IEEE Electron Dev. Lett.*, 20(7):357–359, 1999.
- [4] G. Hobler and S. Selberherr. Monte Carlo simulation of ion implantation into two- and three-dimensional structures. *IEEE Trans. Comp.-Aided Des.*, 8(5):450–459, 1989.
- [5] G. Hobler, A. Lugstein, W. Brezna, and E. Bertagnolli. Simulation of focused ion beam induced damage formation in crystalline silicon. In *Mat. Res. Soc. Symp. Proc.*, volume 792, pages R10.10.1–6. MRS, Warrendale, 2003.
- [6] A. Simionescu and G. Hobler. Two-Dimensional Monte Carlo Simulation of Ion Implantation in Crystalline Silicon Considering Damage Formation. In S. Selberherr, H. Stippel, and E. Strasser, editors, *Simulation of Semiconductor Devices and Processes*, volume 5, pages 361–364. Springer, Wien, 1993.
- [7] G. Hobler and S. Selberherr. Two-dimensional modeling of ion implantation induced point defects. *IEEE Trans. Comp.-Aided Des.*, 7(2):174–180, 1988.
- [8] W. Möller and W. Eckstein. Ion mixing and recoil implantation simulations by means of TRIDYN. *Nucl. Instrum. Meth. Phys. Res. B*, 7–8, Part 2:645–649, 1985.
- [9] A. Simionescu, G. Hobler, S. Bogen, L. Frey, and H. Ryssel. Model for the electronic stopping of channeled ions in silicon around the stopping power maximum. *Nucl. Instrum. Meth. B*, 106:47–50, 1995.
- [10] G. Hobler and C. S. Murthy. Towards a comprehensive model of electronic stopping in amorphous and crystalline silicon. In *2000 Intl. Conf. Ion Implantation Technology Proc.*, pages 209–212, Alpbach, Austria, 2000. IEEE, Piscataway.
- [11] G. Hobler and G. Otto. Status and open problems in modeling as-implanted damage in silicon. *Mat. Sci. Semicond. Proc.*, 6(1–3):1–14, 2003.

- [12] G. Hobler and A. Simionescu. Acceleration of binary collision simulations in crystalline targets using critical angles for ion channeling. *Nucl. Instrum. Meth. B*, 102:24–28, 1995.
- [13] D. Kovač and G. Hobler. Investigation of the impact of defect models on Monte Carlo simulations of RBS/C spectra. *Nucl. Instrum. Meth. B*, 249:776–779, 2006.
- [14] G. Hobler, G. Otto, D. Kovač, L. Palmetshofer, K. Mayerhofer, and K. Piplits. Multi-scale approach for the analysis of channeling profile measurements of ion implantation damage. *Nucl. Instrum. Meth. B*, 228:360–363, 2005.
- [15] G. Otto, D. Kovač, and G. Hobler. Coupled BC/kLMC simulations of the temperature dependence of implant damage formation in silicon. *Nucl. Instrum. Meth. B*, 228:256–259, 2005.
- [16] G. Hobler, K. Nordlund, M. Current, and W. Schustereder. Simulation study of Al channeling in 4H-SiC. In *22nd Intl. Conf. Ion Implantation Technology (IIT)*, pages 247–250, Würzburg, Germany, 2018. IEEE.
- [17] K. Schlueter, K. Nordlund, G. Hobler, M. Balden, F. Granberg, O. Flinck, T. F. da Silva, and R. Neu. Absence of a crystal direction regime in which sputtering corresponds to amorphous material. *Phys. Rev. Lett.*, 125(22):225502, 2020.
- [18] C. Ebm and G. Hobler. Assessment of approximations for efficient topography simulation of ion beam processes: 10 keV Ar on Si. *Nucl. Instrum. Meth. B*, 267:2987–2990, 2009.
- [19] G. Hobler. Assessment of surface potential models by molecular dynamics simulations of atom ejection from (1 0 0)-Si surfaces. *Nucl. Instrum. Meth. B*, 303:165–169, 2013.
- [20] S. J. Lindsey, G. Hobler, D. Maciążek, and Z. Postawa. Simple model of surface roughness for binary collision sputtering simulations. *Nucl. Instrum. Meth. B*, 393:17–21, 2017.
- [21] G. Hobler and D. Kovač. Dynamic binary collision simulation of focused ion beam milling of deep trenches. *Nucl. Instrum. Meth. B*, 269:1609–1613, 2011.
- [22] H. M. Urbassek, M. L. Nietiadi, R. M. Bradley, and G. Hobler. Sputtering of Si_cGe_{1-c} nanospheres. *Phys. Rev. B*, 97(15):155408, 2018.
- [23] R. M. Bradley and G. Hobler. Second order corrections to the sputter yield of a curved surface. *J. Appl. Phys.*, 129(19):194301, 2021.
- [24] G. Hobler. Combined binary collision and continuum mechanics model applied to focused ion beam milling of a silicon membrane. *Nucl. Instrum. Meth. B*, 352:22–26, 2015.

- [25] J. P. Biersack and L. G. Haggmark. A Monte Carlo computer program for the transport of energetic ions in amorphous targets. *Nucl. Instr. Meth.*, 174:257–269, 1980.
- [26] D. S. Gemmell. Channeling and related effects in the motion of charged particles through crystals. *Rev. Mod. Phys.*, 46(1):129–227, 1974.
- [27] G. Hobler. The role of lattice vibrations and interatomic potentials in the simulation of ion implantation. In *NASECODE IX*, pages 55–56. Front Range Press, Boulder, Colorado, 1993.
- [28] G. Hobler, A. Simionescu, L. Palmeshofer, F. Jahnel, R. von Criegern, C. Tian, and G. Stingeder. Verification of models for the simulation of boron implantation into crystalline silicon. *J. Vac. Sci. Technol. B*, 14(1):272–277, 1996.
- [29] G. Hobler and G. Fehlmann. A study of ultra-shallow implanted dopant profiles in silicon using BC and MD simulations. *Radiation Effects and Defects in Solids*, 141(1-4):113–125, 1997.
- [30] G. Hobler, H.-H. Vuong, J. Bevk, A. Agarwal, H.-J. Gossmann, D. C. Jacobson, M. Foad, A. Murrell, and Y. Erokhin. Modeling of ultra-low energy boron implantation in silicon. In *1997 IEDM Techn. Dig.*, pages 489–492. IEEE, Piscataway, 1997.
- [31] G. Hobler and G. Betz. On the useful range of application of molecular dynamics simulations in the recoil interaction approximation. *Nucl. Instrum. Meth. B*, 180(1–4):203–208, 2001.
- [32] M. T. Robinson and I. M. Torrens. Computer simulation of atomic-displacement cascades in solids in the binary-collision approximation. *Phys. Rev.*, B 9(12):5008–5024, 1974.
- [33] K. Gärtner, D. Stock, B. Weber, G. Betz, M. Hautala, G. Hobler, M. Hou, S. Sarite, W. Eckstein, J. J. Jimenez-Rodriguez, A. M. C. Perez-Martin, E. P. Andribet, V. Konoplev, A. Gras-Marti, M. Posselt, M. H. Shapiro, T. A. Tombrello, H. M. Urbassek, H. Hensel, Y. Yamamura, and W. Takeuchi. Round robin computer simulation of ion transmission through thin crystalline layers. *Nucl. Instrum. Meth. B*, 102:183–197, 1995.
- [34] G. Hobler and K. Nordlund. Channeling maps for Si ions in Si: Assessing the binary collision approximation. *Nucl. Instrum. Meth. B*, 449:17–21, 2019.
- [35] J. F. Ziegler, J. P. Biersack, and U. Littmark. *The Stopping and Range of Ions in Solids*. Pergamon Press, New York, 1985.
- [36] G. Hobler and S. Selberherr. Efficient two-dimensional Monte Carlo simulation of ion implantation. In J. J. H. Miller, editor, *NASECODE V*, pages 225–230. Boole Press, Dublin, 1987.

- [37] M. J. Norgett, M. T. Robinson, and I. M. Torrens. A proposed method of calculating displacement dose rates. *Nucl. Eng. Des.*, 33:50–54, 1975.
- [38] G. Hobler. Net point defect concentrations after ion implantation in silicon. In *Process Physics and Modeling in Semiconductor Technology*, pages 509–521. The Electrochem. Soc., Pennington, 1996.
- [39] G. Hobler, A. Simionescu, L. Palmeshofer, C. Tian, and G. Stingeder. Boron channeling implantations in silicon: Modeling of electronic stopping and damage accumulation. *J. Appl. Phys.*, 77(8):3697–3703, 1995.
- [40] S. Herzog, A. Simionescu, G. Hobler, L. Palmeshofer, K. Piplits, and M. Grasserbauer. Monte Carlo simulation of channeled phosphorus implantations in silicon. In *ESSDERC 95*, pages 187–190. Editions Frontieres, Gif sur Yvette Cedex, 1995.
- [41] A. Simionescu, S. Herzog, G. Hobler, R. Schork, J. Lorenz, C. Tian, and G. Stingeder. Modeling of electronic stopping and damage accumulation during arsenic implantation in silicon. *Nucl. Instrum. Meth. B*, 100:483–489, 1995.
- [42] A. Simionescu, G. Hobler, and F. Lau. Monte Carlo simulation of multiple-species ion implantation and its application to the modeling of 0.1 μm PMOS devices. In *Simulation of Semiconductor Devices and Processes, Vol. 6*, pages 484–487. Springer, Wien, 1995.
- [43] A. Simionescu and G. Hobler. Comparison of damage accumulation models for boron implantation in silicon. In *Mat. Res. Soc. Symp. Proc.*, volume 389, pages 221–226. MRS, Pittsburgh, 1995.
- [44] G. Hobler, A. Simionescu, L. Palmeshofer, C. Tian, and G. Stingeder. Boron channeling implantations in silicon: Modeling of electronic stopping and damage accumulation. *J. Appl. Phys.*, 77(8):3697–3703, 1995.
- [45] G. Hobler and G. Otto. Amorphous pocket model for silicon based on molecular dynamics simulations. *Nucl. Instrum. Meth. B*, 206:81–84, 2003.
- [46] T. Henkel, V. Heera, R. Kögler, W. Skorupa, and M. Seibt. Kinetics of ion-beam-induced interfacial amorphization in silicon. *J. Appl. Phys.*, 82(11):5360–5373, 1997.
- [47] Peter Sigmund. Theory of sputtering. I. sputtering yield of amorphous and polycrystalline targets. *Phys. Rev.*, 184(2):383–416, 1969.
- [48] O. S. Oen and M. T. Robinson. Computer studies of the reflection of light ions from solids. *Nucl. Instr. Meth.*, 132:647–653, 1976.
- [49] O. B. Firsov. A qualitative interpretation of the mean electron excitation energy in atomic collisions. *Sov. Phys. JETP*, 36(9)(5):1076–1080, 1959.

- [50] J. Lindhard and M. Scharff. Energy dissipation by ions in the kev region. *Phys. Rev.*, 124(1):128–130, 1961.
- [51] G. Hobler and H. Pötzl. Electronic stopping of channeled ions in silicon. In *Mat. Res. Soc. Symp. Proc.*, volume 279, pages 165–170. MRS, Pittsburgh, 1993.
- [52] G. Hobler, K. K. Bourdelle, and T. Akatsu. Random and channeling stopping power of H in Si below 100 keV. *Nucl. Instrum. Meth. B*, 242:617–619, 2006.
- [53] G. Konac, S. Kalbitzer, C. Klatt, D. Niemann, and R. Stoll. Energy loss and straggling of H and He ions of keV energies in Si. *Nucl. Instrum. Meth. B*, 136–138:159–165, 1998.
- [54] J. H. Barrett. Monte Carlo channeling calculations. *Phys. Rev.*, B 3(5):1527–1547, 1971.
- [55] J.F. Ziegler. SRIM-2013, <http://www.srim.org>, 2013.
- [56] G. Hobler, D. Maciążek, and Z. Postawa. Crater function moments: Role of implanted noble gas atoms. *Phys. Rev. B*, 97(15):155307, 2018.
- [57] G. Hobler, L. Pelaz, and C. S. Rafferty. Continuum treatment of spatial correlation in damage annealing. *Nucl. Instrum. Meth. B*, 153:172–176, 1999.
- [58] G. Hobler and C. S. Murthy. Towards a comprehensive model of electronic stopping in amorphous and crystalline silicon. In *Ion Implantation Technology—2000*, pages 209–212. IEEE, 2000.
- [59] G. Hobler, K. K. Bourdelle, and T. Akatsu. Random and channeling stopping power of H in Si below 100 keV. *Nucl. Instrum. Meth. B*, 242:617–619, 2006.

Full list of publications using IMSIL and its precursors

- [1] G. Hobler and S. Selberherr. Efficient two-dimensional Monte Carlo simulation of ion implantation. In *NASECODE V*, pages 225–230. Boole Press, Dublin, 1987.
- [2] G. Hobler and S. Selberherr. Two-dimensional modeling of ion implantation induced point defects. *IEEE Trans. Comp.-Aided Des.*, 7(2):174–180, 1988.
- [3] G. Hobler and S. Selberherr. Monte Carlo simulation of ion implantation into two- and three-dimensional structures. *IEEE Trans. Comp.-Aided Des.*, 8(5):450–459, 1989.
- [4] G. Hobler, H. Pötzl, L. Palmethofer, R. Schork, J. Lorenz, C. Tian, S. Gara, and G. Stingeder. An empirical model for the electronic stopping of boron in silicon. *COMPEL*, 10(4):323–330, 1991.
- [5] G. Hobler and H. Pötzl. Simulation of two-dimensional implantation profiles with a large concentration range in crystalline silicon using an advanced Monte Carlo method. In *1991 IEDM Techn. Dig.*, pages 693–696. IEEE, Piscataway, 1991.
- [6] G. Hobler, H. Pötzl, L. Gong, and H. Ryssel. Two-dimensional Monte Carlo simulation of boron implantation in crystalline silicon. In *Simulation of Semiconductor Devices and Processes*, volume 4, pages 389–398. Hartung Gorre, Konstanz, 1991.
- [7] G. Hobler and H. Pötzl. The effect of a screening oxide on ion implantation studied by Monte Carlo simulations. *COMPEL*, 11(4):403–411, 1992.
- [8] G. Hobler and H. Pötzl. Electronic stopping of channeled ions in silicon. In *Mat. Res. Soc. Symp. Proc.*, volume 279, pages 165–170. MRS, Pittsburgh, 1993.
- [9] G. Hobler. The role of lattice vibrations and interatomic potentials in the simulation of ion implantation. In *NASECODE IX*, pages 55–56. Front Range Press, Boulder, Colorado, 1993.
- [10] A. Simionescu, G. Hobler, and R. von Criegern. Monte Carlo simulation of ion implantation into two-dimensional structures and its application to the prediction of

- lateral SIMS results. In *ESSDERC 94*, pages 97–100. Editions Frontieres, Gif-sur-Yvette Cedex, 1994.
- [11] K. Gärtner, D. Stock, B. Weber, G. Betz, M. Hautala, G. Hobler, M. Hou, S. Sarite, W. Eckstein, J. J. Jimenez-Rodriguez, A. M. C. Perez-Martin, E. P. Andribet, V. Konoplev, A. Gras-Marti, M. Posselt, M. H. Shapiro, T. A. Tombrello, H. M. Urbassek, H. Hensel, Y. Yamamura, and W. Takeuchi. Round robin computer simulation of ion transmission through thin crystalline layers. *Nucl. Instrum. Meth. B*, 102:183–197, 1995.
- [12] S. Herzog, A. Simionescu, G. Hobler, L. Palmethofer, K. Piplits, and M. Grasserbauer. Monte Carlo simulation of channeled phosphorus implantations in silicon. In *ESSDERC 95*, pages 187–190. Editions Frontieres, Gif sur Yvette Cedex, 1995.
- [13] G. Hobler, A. Simionescu, L. Palmethofer, C. Tian, and G. Stingeder. Boron channeling implantations in silicon: Modeling of electronic stopping and damage accumulation. *J. Appl. Phys.*, 77(8):3697–3703, 1995.
- [14] G. Hobler. Monte Carlo simulation of two-dimensional implanted dopant distributions at mask edges. *Nucl. Instrum. Meth. B*, 96:155–162, 1995.
- [15] A. Simionescu and G. Hobler. Comparison of damage accumulation models for boron implantation in silicon. In *Mat. Res. Soc. Symp. Proc.*, volume 389, pages 221–226. MRS, Pittsburgh, 1995.
- [16] A. Simionescu, S. Herzog, G. Hobler, R. Schork, J. Lorenz, C. Tian, and G. Stingeder. Modeling of electronic stopping and damage accumulation during arsenic implantation in silicon. *Nucl. Instrum. Meth. B*, 100:483–489, 1995.
- [17] A. Simionescu, G. Hobler, S. Bogen, L. Frey, and H. Ryssel. Model for the electronic stopping of channeled ions in silicon around the stopping power maximum. *Nucl. Instrum. Meth. B*, 106:47–50, 1995.
- [18] A. Simionescu, G. Hobler, and F. Lau. Monte Carlo simulation of multiple-species ion implantation and its application to the modeling of 0.1 μm PMOS devices. In *Simulation of Semiconductor Devices and Processes, Vol. 6*, pages 484–487. Springer, Wien, 1995.
- [19] G. Hobler and A. Simionescu. Acceleration of binary collision simulations in crystalline targets using critical angles for ion channeling. *Nucl. Instrum. Meth. B*, 102:24–28, 1995.
- [20] G. Hobler. Critical angles and low-energy limits to ion channeling in silicon. *Radiat. Eff. Def. Sol.*, 139:21–85, 1996.
- [21] G. Hobler. Net point defect concentrations after ion implantation in silicon. In *Process Physics and Modeling in Semiconductor Technology*, pages 509–521. The Electrochem. Soc., Pennington, 1996.

- [22] G. Hobler. Theoretical estimate of the low-energy limit to ion channeling. *Nucl. Instrum. Meth. B*, 115:323–327, 1996.
- [23] G. Hobler, A. Simionescu, L. Palmetshofer, F. Jahnel, R. von Criegern, C. Tian, and G. Stingeder. Verification of models for the simulation of boron implantation into crystalline silicon. *J. Vac. Sci. Technol. B*, 14(1):272–277, 1996.
- [24] G. Hobler, H.-H. Vuong, J. Bevk, A. Agarwal, H.-J. Gossmann, D. C. Jacobson, M. Foad, A. Murrell, and Y. Erokhin. Modeling of ultra-low energy boron implantation in silicon. In *1997 IEDM Techn. Dig.*, pages 489–492. IEEE, Piscataway, 1997.
- [25] G. Hobler and G. Fehlmann. A study of ultra-shallow implanted dopant profiles in silicon using BC and MD simulations. *Radiation Effects and Defects in Solids*, 141(1-4):113–125, 1997.
- [26] A. Y. Nikulin, A. W. Stevenson, H. Hashizume, D. Cookson, G. Hobler, and S.W. Wilkins. Model-independent determination of 2D strain distribution in ion implanted silicon crystals from x-ray diffraction data. *Semicond. Sci. Technol.*, 12(3):350–354, 1997.
- [27] R. von Criegern, F. Jahnel, R. Lange-Gieseler, P. Pearson, G. Hobler, and A. Simionescu. Verification of “lateral secondary ion mass spectrometry” as a method for measuring lateral dopant dose distributions in microelectronics test structures. *J. Vac. Sci. Technol. B*, 16(1):386–393, 1998.
- [28] G. Hobler, J. Bevk, and A. Agarwal. Channeling of low-energy implanted ions through the poly-Si gate. *IEEE Electron Dev. Lett.*, 20(7):357–359, 1999.
- [29] G. Hobler, L. Pelaz, and C. S. Rafferty. Continuum treatment of spatial correlation in damage annealing. *Nucl. Instrum. Meth. B*, 153:172–176, 1999.
- [30] G. Hobler and C. S. Rafferty. Modeling of {311} Defects. In *Mat. Res. Soc. Sym. Proc.*, volume 568, pages 123–134. MRS Warrendale, 1999.
- [31] G. Hobler, L. Pelaz, and C. S. Rafferty. Dose, energy, and ion species dependence of the effective plus factor for transient enhanced diffusion. *J. Electrochem. Soc.*, 147(9):3494–3501, 2000.
- [32] G. Hobler and C. S. Murthy. Towards a comprehensive model of electronic stopping in amorphous and crystalline silicon. In *2000 Intl. Conf. Ion Implantation Technology Proc.*, pages 209–212, Alpbach, Austria, 2000. IEEE, Piscataway.
- [33] W. Boxleitner and G. Hobler. FIBSIM — Dynamic Monte Carlo simulation of compositional and topography changes caused by focused ion beam milling. *Nucl. Instrum. Meth. B*, 180:125–129, 2001.

- [34] W. Boxleitner, G. Hobler, V. Klüppel, and H. Cerva. Dynamic simulation of topography evolution and damage formation in TEM sample preparation using focused ion beams. *Nucl. Instrum. Meth. B*, 175–177:102–107, 2001.
- [35] G. Hobler and V. Moroz. Initial conditions for transient enhanced diffusion: Beyond the plus-factor approach. In *Proc. SISPAD 2001*, pages 34–37. Springer, Wien, 2001.
- [36] G. Hobler and G. Betz. On the useful range of application of molecular dynamics simulations in the recoil interaction approximation. *Nucl. Instrum. Meth. B*, 180(1–4):203–208, 2001.
- [37] L. Palmetshofer, M. Gritsch, and G. Hobler. Range of ion-implanted rare earth elements in Si and SiO₂. *Mater. Sci. Eng. B*, 81:83–85, 2001.
- [38] G. Hobler and G. Otto. Amorphous pocket model for silicon based on molecular dynamics simulations. *Nucl. Instrum. Meth. B*, 206:81–84, 2003.
- [39] G. Hobler, A. Lugstein, W. Brezna, and E. Bertagnolli. Simulation of focused ion beam induced damage formation in crystalline silicon. In *Mat. Res. Soc. Symp. Proc.*, volume 792, pages R10.10.1–6. MRS, Warrendale, 2003.
- [40] G. Hobler and G. Otto. Status and open problems in modeling as-implanted damage in silicon. *Mat. Sci. Semicond. Proc.*, 6(1–3):1–14, 2003.
- [41] A. Lugstein, W. Brezna, G. Hobler, and E. Bertagnolli. Method to characterize the 3D-distribution of focused ion beam induced damage in silicon after 50 keV Ga⁺ irradiation. *J. Vac. Sci. Technol. A*, 21(5):1644–1648, 2003.
- [42] G. Otto and G. Hobler. Coupled kinetic Monte Carlo and molecular dynamics simulations of implant damage accumulation in silicon. In *Mat. Res. Soc. Symp. Proc.*, volume 792, pages R6.5.1–6. MRS, Warrendale, 2003.
- [43] G. Otto, G. Hobler, L. Palmetshofer, K. Mayerhofer, K. Piplits, and H. Hutter. Dose-rate dependence of damage formation in Si by N implantation as determined from channeling profile measurements. *Nucl. Instrum. Meth. B*, 242:667–669, 2005.
- [44] D. Kovač, G. Otto, and G. Hobler. Modeling of amorphous pocket formation in silicon by numerical solution of the heat transport equation. *Nucl. Instrum. Meth. B*, 228:226–229, 2005.
- [45] G. Hobler, G. Otto, D. Kovač, L. Palmetshofer, K. Mayerhofer, and K. Piplits. Multi-scale approach for the analysis of channeling profile measurements of ion implantation damage. *Nucl. Instrum. Meth. B*, 228:360–363, 2005.
- [46] G. Otto, D. Kovač, and G. Hobler. Coupled BC/kLMC simulations of the temperature dependence of implant damage formation in silicon. *Nucl. Instrum. Meth. B*, 228:256–259, 2005.

- [47] T. Zahel, G. Otto, and G. Hobler. Atomistic simulation of the isotope effect on defect formation in H/D-implanted Si. In *Silicon-on-Insulator Technology and Devices XII*, pages 179–184. The Electrochem. Soc., Pennington, 2005.
- [48] G. Hobler, K. K. Bourdelle, and T. Akatsu. Random and channeling stopping power of H in Si below 100 keV. *Nucl. Instrum. Meth. B*, 242:617–619, 2006.
- [49] D. Kovač and G. Hobler. Investigation of the impact of defect models on Monte Carlo simulations of RBS/C spectra. *Nucl. Instrum. Meth. B*, 249:776–779, 2006.
- [50] G. Otto, G. Hobler, P. Pongratz, and L. Palmetshofer. Is there an influence of ion-beam-induced interfacial amorphization on the a/c-interface depth in silicon at common implantation energies? *Nucl. Instrum. Meth. B*, 253:227–231, 2006.
- [51] H.-B. Kim, G. Hobler, A. Lugstein, and E. Bertagnolli. Simulation of ion beam induced micro/nano fabrication. *J. Micromech. Microeng.*, 17:1178–1183, 2007.
- [52] G. Otto, G. Hobler, L. Palmetshofer, and P. Pongratz. Amorphous pockets in Si: Comparison of coupled molecular dynamics and TEM image contrast simulations with experimental results. *Nucl. Instrum. Meth. B*, 255:105–109, 2007.
- [53] C. Ebm and G. Hobler. Simulation of ion-beam induced etching and deposition using a non-local recoil-based algorithm. In *Mater. Res. Soc. Symp. Proc.*, volume 1181, pages DD03–04, 2009.
- [54] C. Ebm and G. Hobler. Assessment of approximations for efficient topography simulation of ion beam processes: 10 keV Ar on Si. *Nucl. Instrum. Meth. B*, 267:2987–2990, 2009.
- [55] C. Ebm, E. Platzgummer, G. Hobler, A. Koeck, R. Hainberger, M. Wellenzohn, S. Eder-Kapl, P. Joechl, M. Kuemmel, R. Reitinger, F. Letzkus, M. Irmscher, G. Gross, and H. Loeschner. Ion multi-beam nanopatterning for photonic applications: experiments and simulations, including study of precursor gas induced etching and deposition. *J. Vac. Sci. Technol. B*, 27(6):2668–2673, 2009.
- [56] D. Kovač and G. Hobler. Amorphous pocket model based on the modified heat transport equation and local lattice collapse. *Nucl. Instrum. Meth. B*, 267:1229–1231, 2009.
- [57] C. Ebm, G. Hobler, S. Waid, and H. D. Wanzenboeck. Quantitative simulation of ion-beam induced deposition of nanostructures. *J. Vac. Sci. Technol. B*, 29(1):011031, 2011.
- [58] G. Hobler and D. Kovač. Dynamic binary collision simulation of focused ion beam milling of deep trenches. *Nucl. Instrum. Meth. B*, 269:1609–1613, 2011.

- [59] H.-B. Kim, G. Hobler, A. Steiger, A. Lugstein, E. Bertagnolli, E. Platzgummer, and H. Loeschner. Sputter-redeposition method for the fabrication of automatically sealed micro/nanochannel using FIBs. *Int. J. Prec. Eng. Manuf.*, 12(5):893–898, 2011.
- [60] H.-B. Kim and G. Hobler. Simulation of Focused Ion Beam Milling. In *Nanofabrication Using Focused Ion and Electron Beams: Principles and Applications*, pages 226–247. Oxford University Press, 2012.
- [61] S. Lindsey and G. Hobler. The significance of redeposition and backscattering in nanostructure formation by focused ion beams. *Nucl. Instrum. Meth. B*, 282(0):12–16, 2012.
- [62] G. Hobler. Assessment of surface potential models by molecular dynamics simulations of atom ejection from (1 0 0)-Si surfaces. *Nucl. Instrum. Meth. B*, 303:165–169, 2013.
- [63] S. Lindsey and G. Hobler. Sputtering of silicon at glancing incidence. *Nucl. Instrum. Meth. B*, 303:142–147, 2013.
- [64] S. Lindsey, S. Waid, G. Hobler, H. D. Wanzenböck, and E. Bertagnolli. Inverse modeling of FIB milling by dose profile optimization. *Nucl. Instrum. Meth. B*, 341:77–83, 2014.
- [65] G. Hobler. Combined binary collision and continuum mechanics model applied to focused ion beam milling of a silicon membrane. *Nucl. Instrum. Meth. B*, 352:22–26, 2015.
- [66] G. Hobler, R. M. Bradley, and H. M. Urbassek. Probing the limitations of Sigmund’s model of spatially resolved sputtering using Monte Carlo simulations. *Phys. Rev. B*, 93(20):205443, 2016.
- [67] G. Hobler, M. L. Nietiadi, R. M. Bradley, and H. M. Urbassek. Sputtering of silicon membranes with nanoscale thickness. *J. Appl. Phys.*, 119(24):245105, 2016.
- [68] S. J. Lindsey, G. Hobler, D. Maciążek, and Z. Postawa. Simple model of surface roughness for binary collision sputtering simulations. *Nucl. Instrum. Meth. B*, 393:17–21, 2017.
- [69] G. Hobler, K. Nordlund, M. Current, and W. Schustereder. Simulation study of Al channeling in 4H-SiC. In *22nd Intl. Conf. Ion Implantation Technology (IIT)*, pages 247–250, Würzburg, Germany, 2018. IEEE.
- [70] G. Hobler, D. Maciążek, and Z. Postawa. Crater function moments: Role of implanted noble gas atoms. *Phys. Rev. B*, 97(15):155307, 2018.
- [71] M. Current, G. Hobler, Y. Kawasaki, and M. Sugitani. Channeled MeV B, P and As profiles in Si(100): Monte Carlo models and SIMS. In *22nd International Conference on Ion Implantation Technology (IIT)*, pages 251–254, Würzburg, 2018. IEEE.

- [72] H. M. Urbassek, M. L. Nietiadi, R. M. Bradley, and G. Hobler. Sputtering of $\text{Si}_c\text{Ge}_{1-c}$ nanospheres. *Phys. Rev. B*, 97(15):155408, 2018.
- [73] G. Hobler and K. Nordlund. Channeling maps for Si ions in Si: Assessing the binary collision approximation. *Nucl. Instrum. Meth. B*, 449:17–21, 2019.
- [74] G. Hobler, D. Maciążek, and Z. Postawa. Ion bombardment induced atom redistribution in amorphous targets: MD versus BCA. *Nucl. Instrum. Meth. B*, 447:30–33, 2019.
- [75] M. Current, Y. Kawasaki, and G. Hobler. Aspects of highly-channeled MeV implants of dopants in Si(100). In *Intl. Workshop on Junction Technology (IWJT19)*, Uji, Japan, 2019.
- [76] K. Schlueter, K. Nordlund, G. Hobler, M. Balden, F. Granberg, O. Flinck, T. F. da Silva, and R. Neu. Absence of a crystal direction regime in which sputtering corresponds to amorphous material. *Phys. Rev. Lett.*, 125(22):225502, 2020.
- [77] R. M. Bradley and G. Hobler. Second order corrections to the sputter yield of a curved surface. *J. Appl. Phys.*, 129(19):194301, 2021.