

IMSIL 2024-03+

G. Hobler

TU Wien, AUSTRIA

Release Version
(printed January 16, 2025)

Contents

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 January 16, 2025.

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 [?]. The target may either be a 1-D layered structure defined by the interface positions of the materials [?, ?], or a 2-D structure composed of regions defined by arbitrary polygons [?, ?]. 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 [?], 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 [?], to obtain the distribution of atoms recoiled from one target region into another [?], 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 [?, ?, ?, ?], another unique feature of IMSIL is its physics-based model of extended free flight paths in amorphous targets [?], 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 [?], and to use a deterministic model of damage as an alternative to the standard statistic model [?, ?]. 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 [?]. More recently the possibility to read in arbitrary orthogonal unit cells and the fcc, bcc, and sc lattice were implemented [?]. Also, a polycrystalline mode has been implemented [?] 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) [?, ?] and of a surface roughness model that provides more realistic sputter yields for glancing incidence of the ions [?]. The introduction of “virtual recoils” [?] 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 [?, ?]. 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 [?, ?], 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.

The first parallel version of IMSIL was produced in 2023, first using Fortran coarrays and then using OpenMP. One advantage of OpenMP is the possibility to still compile a sequential code which can more easily be run on other computers than the one the compilation was done.

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** E 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 the other. 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** M 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 the nominal beam direction (reference direction) and, optionally, by a beam divergence. The reference direction \hat{z}' is defined by tilt angles θ_x and θ_y and a rotation angle φ relative to the target coordinate system $(\hat{x}, \hat{y}, \hat{z})$, see Fig. ???. To be more precise, the beam coordinate system $(\hat{x}', \hat{y}', \hat{z}')$ is obtained from the target coordinate system $(\hat{x}, \hat{y}, \hat{z})$ by first rotating the system about the z axis by an angle φ , then about the new x axis by θ_x ($\theta_x = 0$ in Fig. ??), and finally about the new y axis by θ_y . In symbols:

$$(\hat{x}, \hat{y}, \hat{z}) \rightarrow (\hat{x}', \hat{y}', \hat{z}') : \mathbf{R}_z(\varphi) \rightarrow \mathbf{R}_x(\theta_x) \rightarrow \mathbf{R}_y(\theta_y) \quad (2.1)$$

Note that positive angles for rotations about the z axis are defined so that the positive x axis is rotated towards the positive y axis, for the x axis so that the positive y axis is rotated towards the positive z axis, and for the y axis so that the positive z axis is rotated towards the positive x axis. The corresponding rotation matrix is

$$\mathbf{R} = \mathbf{R}_y(\theta_y)\mathbf{R}_x(\theta_x)\mathbf{R}_z(\varphi) \quad (2.2)$$

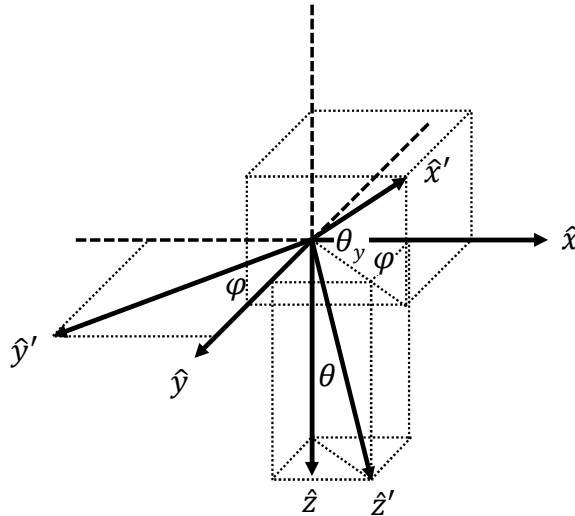


Figure 2.1: Definition of the beam coordinate system $(\hat{x}', \hat{y}', \hat{z}')$ and the target coordinate system $(\hat{x}, \hat{y}, \hat{z})$ by tilt angle θ_y and rotation angle φ .

Constructing the beam coordinate system from the target coordinate system is convenient for theoretical investigations. In practice, the beam coordinate system is usually fixed in space, and the target is rotated. It is therefore more convenient to construct the target coordinate system from the beam coordinate system. To obtain the same relative orientations of the coordinate systems, the order of rotations has to be reversed, and the rotations have to be performed in opposite directions. In symbols:

$$(\hat{x}', \hat{y}', \hat{z}') \rightarrow (\hat{x}, \hat{y}, \hat{z}) : \mathbf{R}_y(-\theta_y) \rightarrow \mathbf{R}_x(-\theta_x) \rightarrow \mathbf{R}_z(-\varphi) \quad (2.3)$$

Using an axis of the rotated coordinate system for the next rotation defines “intrinsic” rotations. In some systems, rotations are always performed about axes defined in the beam coordinate system. Such chained rotations are called “extrinsic”. It can be shown that extrinsic rotations are described by the same rotation matrix as intrinsic rotations (Eq. ??), if the rotations are interpreted as rotations of the target coordinate system relative to the beam coordinate system:

$$(\hat{x}', \hat{y}', \hat{z}') \rightarrow (\hat{x}, \hat{y}, \hat{z}) : \mathbf{R}_z(\varphi) \rightarrow \mathbf{R}_x(\theta_x) \rightarrow \mathbf{R}_y(\theta_y). \quad (2.4)$$

While the individual rotations describe the rotation of the target coordinate system relative to the beam coordinate system, the rotation matrix \mathbf{R} describes the transformation from the target coordinate system to the beam coordinate system like in the “intrinsic” rotations case. Therefore, the user need not specify whether the rotations are extrinsic or intrinsic.

Without beam divergence, the initial directions of all ions coincide with the reference direction \hat{z}' (z axis of the beam coordinate system, cf. Fig. ??). Considering beam divergence means to randomly select the ion’s direction vector on the unit sphere in the

beam coordinate system so that the probability per solid angle follows a given distribution centered around the reference direction. The beam divergence models may be chosen from two options: isotropic within a maximum rotation angle, and Gaussian with given standard deviations in \hat{x}' and \hat{y}' direction.

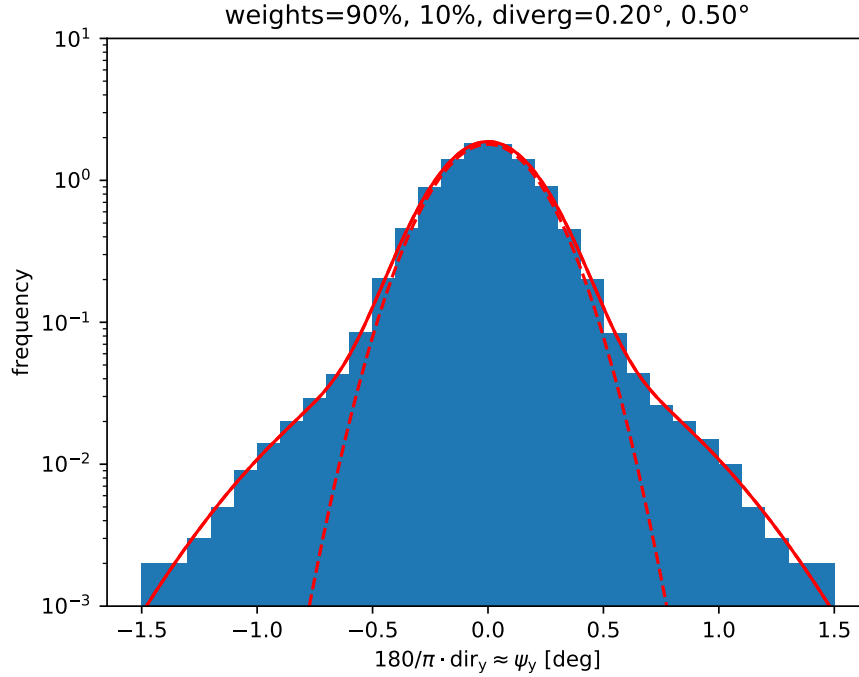


Figure 2.2: Beam profile composed of two Gaussian functions (red line) and histogram of angles ψ_y generated by IMSIL.

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 weight from each of the partial beams. An example for the superposition of two Gaussian beam divergence distributions is shown in Fig. ?? . Below is an excerpt from an IMSIL input file that specifies this 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 variable, 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 3.3, 3.4, and 3.5.

| parameter | IMSIL name | to be specified in record |
|--|------------|---------------------------|
| internal starting point flag | LIINIT | &IONS |
| ion name | NAME | &IONS |
| energy E | ENERGY | &IONS |
| atom mass M | MASS | &ATOMS |
| reference starting point | XINIT | &IONS |
| | YINIT | &IONS |
| | ZINIT | &IONS |
| beam FWHM (FWHM _{<i>x</i>} , FWHM _{<i>y</i>}) | FWHM | &IONS |
| tilt angle θ_x | TILTX | &IONS |
| tilt angle θ_y | TILTY=TILT | &IONS |
| rotation angle φ | ROTATE | &IONS |
| random rotation flag | RANROT | &IONS |
| beam divergence model | MODDIV | &IONS |
| beam divergence (ψ_x, ψ_y) | 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 3-D geometries for static simulations, and 1-D geometries for dynamic simulations. Static simulations may also use rotational symmetry to simplify geometry definition and/or accelerate simulation.

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. For further details see Section ??.

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 4.9). 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 signed distance function array for each region. The distance values are defined on a regular 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 [?].

2-D and arbitrary 3-D geometries may also be defined by signed distance function (SDF) values on a regular grid, provided as external input on files. An SDF specifies the distance of the grid points from the nearest point on the boundary of the region under consideration. It is given a negative sign when the grid point is outside the region. An SDF must also be provided for the whole simulations domain (= the union of all regions) including a rim of thickness at least equal to the maximum free flight path. (?) All region SDFs must provide values on the whole simulation domain. Currently, the user is responsible for the consistency of the SDFs.

Finally, a simple **surface roughness** model is provided for both 1-D and 2-D simulations. This so-called density gradient model [?] 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 | &GEOMETRY |
| cylindrical symmetry flag | LCYLX | &GEOMETRY |
| | LCYLY | &GEOMETRY |
| | LCYLZ | &GEOMETRY |
| spherical symmetry flag | LSPHER | &GEOMETRY |
| position of symmetry line/center | CENTER | &GEOMETRY |
| 2-D geometry file | GEOMFILE | &GEOMETRY |
| polygon vertices | POS | &GEOMETRY |
| polygon connectivity | POINTS | &GEOMETRY |
| periodicity flag | LPERX | &GEOMETRY |
| periodic boundary positions | XPER | &GEOMETRY |
| box width of distance grid | WBOX2 | &GEOMETRY |
| maximum size of distance grid | NBOX2 | &GEOMETRY |
| signed distance function file | SDFFILE | &GEOMETRY |
| thickness of roughness layer | WROUGH | &GEOMETRY |

2.4 Dynamic Target Updates

In a **static** simulation, the target presents itself to all incoming ions as specified in the input file and, optionally, any file read during setup (histogram, geometry, or cell file).

In a **dynamic** simulation, the target is updated according to the changes caused by the simulated collision cascades. IMSIL supports two kinds of dynamic simulation modes: First (“damage accumulation”), the damage to crystalline targets may be taken into account for subsequent trajectories. In this way, the suppression of channeling by damage accumulation may be described. Also, recombination of newly generated defects with damage from previous collision cascades can be taken into account. Second (“dynamic simulation”), it is possible to predict the evolution of the target geometry from the changes in the atom densities of the target. In the version described in this manual, only 1-D targets can be updated in this way.

The updates can be done continuously or with a certain frequency. In the sequential version of IMSIL, damage and atom densities are, by default, updated as soon as they occur. In the parallel version this would lead to excessive communication between the threads. Therefore, the user has to choose the frequency of these updates in terms of the number of simulated ions between updates. Too frequent updates lead to penalties in execution time, while too infrequent updates may affect the results. In case of geometry updates, it is also possible to specify that updates be performed only when cell densities have changed more than a threshold. When a geometry update is performed, also the grid for the atom densities is adjusted so that the grid spacings remain approximately the same.

| parameter | IMSIL name | to be specified in record |
|---|------------|---------------------------|
| histogram file | HISFILE | &SETUP |
| geometry file | GEOMFILE | &GEOMETRY |
| cell file | CELLFILE | &SETUP |
| take damage accumulation into account | LDAMDYN | &SETUP |
| dynamic simulation mode | LDYN | &SETUP |
| number of ions for histogram updates | NIONHIS | &SETUP |
| number of ions for geometry updates | NIONGEOM | &SETUP |
| refined damage model | LDAM | &DAMAGE |
| threshold for geometry update | FRACUPD | &GEOMETRY |
| fixed coordinate in 1-D geometry update | POSFIX | &GEOMETRY |

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 4.

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 [3], 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 [?]. 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 [3], 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. (2.1). 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 [6]. A value of 490 K is recommended for the Debye temperature of Si [?, ?]. The values of the Debye temperatures for all elements are tabulated within IMSIL and appropriate vibrational amplitudes are calculated using the Debye function [6].

It has been shown that “simultaneous collisions” are not beneficial for dopant profile calculation [?, ?, ?]. “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} [29]. 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 [?] and other comparisons with MD simulations [?, ?].

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

Interatomic potentials for binary collision simulation are usually written as screened Coulomb potentials

$$V(r) = \frac{1}{4\pi\epsilon_0} \frac{Z_1 Z_2 e^2}{r} \cdot \Phi\left(\frac{r}{a_{12}}\right). \quad (3.2)$$

Z_1 and Z_2 denote the atomic numbers of the ion and the target atom, respectively, r the interatomic separation, and $e^2/(4\pi\epsilon_0) = 14.4 \text{ eV}\text{\AA}$. The screening function $\Phi(R)$ can either be read from a file or can be chosen as the sum of exponentials with an optional linear term

$$\Phi(R) = \sum_{i=1}^n a_i \cdot \exp(-b_i R) \left[+c + d \frac{R}{R_{\max}} \right]. \quad (3.3)$$

The screening length a_{12} is chosen according to

$$a_{12} = \frac{0.468 \text{ \AA}}{(Z_1^p + Z_2^p)^q} \quad (3.4)$$

The predefined values for the parameters a_i , b_i , p , and q are listed in Tab. ???. Alternatively, a_i , b_i , and optionally R_{\max} can be read from a file. If R_{\max} is given, then c and d are calculated so that $\Phi(0) = 1$ and $\Phi(R_{\max}) = 0$.

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 . For the ZBL screening function, θ may be calculated by bicubic interpolation in a table of $\cot(\theta/2)$ [16]. Such tables may be set up, in principle, for any interatomic potential. The scattering table file corresponding to the universal ZBL potential is called **SCATTAB**. Tables for some of the pair-specific potentials are also provided with the program.¹

¹The 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) [37].

Table 3.1: Predefined parameters for the screening functions and screening lengths

| Screening functions (Eq. ??) | | | |
|------------------------------|----------|----------|-------------|
| | ZBL [37] | KrC [?] | Moliere [?] |
| a_1 | 0.18175 | 0.190945 | 0.35 |
| a_2 | 0.50986 | 0.473674 | 0.55 |
| a_3 | 0.28022 | 0.335381 | 0.10 |
| a_4 | 0.02817 | | |
| b_1 | 3.19980 | 0.278544 | 0.3 |
| b_2 | 0.94229 | 0.637174 | 1.2 |
| b_3 | 0.40290 | 1.919249 | 6.0 |
| b_4 | 0.20162 | | |
| a_{12} (default) | ZBL | Firsov | Firsov |

| Screening lengths (Eq. ??) | | | |
|----------------------------|----------|------------|--------------|
| | ZBL [37] | Firsov [?] | Lindhard [?] |
| p | 0.23 | 1/2 | 2/3 |
| q | 1 | 2/3 | 1/2 |

Alternatively, θ can be determined by numerical integration using Gauss-Legendre or Gauss-Mehler quadrature. This allows to use screening functions defined numerically or with parameters other than according to ZBL. Tables for the screening functions of all pair-specific ZBL or DMol [?] 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 |
| numerical quadrature flag | QUAD | &SNPAR |
| scattering table file | SCATFILE | &SNPAR |
| screening coefficients | COEFFILE | &SNPAR |
| screening function table | SCRFILE | &SNPAR |
| screening length model | SCRLEN | &SNPAR |

The formats of the files SCATFILE, COEFFILE, and SCRFILE are described in Section ??.

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 [26] 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 [?]:

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

$$\Delta n_{\text{I}} = -1 \quad \Delta n_{\text{V}} = 0 \quad (3.5)$$

- (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).

$$\Delta n_{\text{V}} = f_{\text{rec}}(1 - kN_{\text{I}}) \quad \Delta n_{\text{I}} = \Delta n_{\text{V}} - 1 \quad (3.6)$$

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 $1 - kN_{\text{V}}$), 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.

$$\Delta n_{\text{I}} = 1 - kN_{\text{V}} \quad \Delta n_{\text{V}} = \Delta n_{\text{I}} - 1 \quad (3.7)$$

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. (2.5) and (2.6) are replaced by

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

and

$$\Delta n_I = (f_{\text{rg}} + (1 - f_{\text{rg}})f_{\text{rec}})(1 - kN_V) \quad \Delta n_V = \Delta n_I - 1 \quad (3.9)$$

respectively. Note that these equations reduce to Eqs. (2.5) and (2.6) 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.²

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 [?], P [?, ?], As [?, ?], and BF_2 [?, ?] in Si. Further investigations of the damage model are reported in [?, ?]. 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 [?]. 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 noted, however, that one should revert to the values given in [19] if the dose dependence of [110] channeling is investigated.

The model may also be used in a simplified version that does not consider defect recombination with previous cascades, corresponding to $k = 0$ and/or $N_I = N_V = 0$. It is also possible that the damage distributions are only updated after a certain number of cascades.

As an alternative in case of true recombination ($f_{\text{rec}} < 1$), the capture radius model may be used [?]. 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 [?]: 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

²The 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.

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 [?] can be activated. In this model, existing amorphous zones grow with a rate proportional to nuclear energy deposition.

In addition to being recorded, damage to crystalline regions can also be considered in the selection of collision partners.

| parameter | IMSIL name | to be specified in record |
|--|------------|---------------------------|
| flag for considering defect recombination with previous cascades and using damage in crystalline regions | LDAMDYN | &SETUP |
| damage update interval | NIONHIS | &SETUP |
| refined damage model flag | LDAM | &DAMAGE |
| 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 [?] 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 [?] 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 power S_e of a **random** (amorphous) material for a projectile is defined by

$$\left(\frac{dE}{dx}\right)_e = N \cdot S_e \quad (3.10)$$

where $(dE/dx)_e$ denotes the energy loss per path length due to electronic stopping, and N the atomic density (number of atoms per volume). For compounds, Bragg's rule [?] is used, which says that the stopping power S_e of the compound is obtained by a linear superposition of the stopping powers S_{ei} of the elemental materials of atom species i

$$\left(\frac{dE}{dx}\right)_e = \sum_i N_i \cdot S_{ei} \quad (3.11)$$

N_i denotes the atomic density of species i in the compound.

IMSIL provides several choices for S_e as a function of projectile energy E .³ SEMODEL='Lindhard' combines a generalization of Lindhard's stopping power

$$S_{eL} = k_{\text{corr}} \cdot k_L \cdot E^p \quad (3.12)$$

and the slightly modified Bethe-Bloch stopping power [3]

$$S_{eB} = \frac{B}{E_b} \cdot \ln \left(E_b + c_0 + \frac{c_1}{E_b} \right). \quad (3.13)$$

With $p = 0.5$, Eq. (2.15) is Lindhard's model [?], where electronic stopping is proportional to the projectile's velocity. k_L is a factor given in Ref. [?], which depends on the projectile and target atom species. k_{corr} and p can be used to adjust the Lindhard model to experimental data. In Eq. (2.17), B , c_0 , and c_1 are ion- and target-dependent prefactor, and E_b is proportional to the projectile energy E (for details see [3]). The two stopping powers are combined by [3, ?]

$$S_e = \left(S_{eL}^{-c} + S_{eB}^{-c} \right)^{-1/c} \quad (3.14)$$

so that $S_e \approx S_{eL}$ at low energies and $S_e \approx S_{eB}$ at high energies. c can be used to adjust the stopping power around the stopping power maximum, i.e., at energies where $S_{eL} \approx S_{eB}$.

The combined Lindhard and Bethe-Bloch model has been fitted for many projectiles in Si [14] and a few other cases. The parameters are provided as default values. Where such default values exist, the combined Lindhard and Bethe-Bloch model is recommended.

Where no fitted parameters of the combined Lindhard and Bethe-Bloch model exist, SRIM-2013 [21] stopping powers (SEMODEL='SRIM') are recommended. This model evaluates the electronic stopping power by interpolation in tables for all combinations of

³In the following, we drop the index i , but we mean the electronic stopping power of an elemental target.

projectile and target atoms species ($Z = 1 \dots 92$, $E \leq 2$ GeV).⁴ Alternatively, the 1985 version of the TRIM stopping powers [37] can be used (`SEMODEL='ZBL'`). This option is kept for compatibility and is generally not recommended. Another unrecommended option for Si targets is the model by Konac, Klatt, and Kalbitzer [?] (`SEMODEL='KKK'`).

A last option for an electronic stopping model is reading in a table of energy losses $(dE/dx)_e$ versus energy E (`SEMODEL='table'` and `SEFILE`). Such a table can be generated, e.g., using SRIM [21] and commenting the header lines. For the format of the `SEFILE` expected by IMSIL see Section ?? . This option is recommended for obtaining accurate stopping powers of compounds containing light elements [?].

The electronic stopping of channeled ions is reduced as compared to 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

$$\Delta E_e^{\text{nl}} = N \cdot S_e \cdot L \cdot \left[x^{\text{nl}} + x^{\text{loc}} \cdot \left(1 + \frac{p_{\text{max}}}{a} \right) \cdot \exp \left\{ -\frac{p_{\text{max}}}{a} \right\} \right] \quad (3.15)$$

and a part which depends on the impact parameter p

$$\Delta E_e^{\text{loc}} = x^{\text{loc}} \cdot \frac{S_e}{2\pi a^2} \cdot \exp \left\{ -\frac{p}{a} \right\} \quad (3.16)$$

where

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

N denotes the atomic density of the target, and $0 \leq x^{\text{nl}} \leq 1$. For $x^{\text{nl}} = 0$ and a certain choice of a , this is the model of Oen and Robinson [27]. Alternatively, the impact parameter dependence can be chosen by a generalized Firsov model [4], in which case the model reads

$$\Delta E_e^{\text{nl}} = N \cdot S_e \cdot L \cdot \left[x^{\text{nl}} + x^{\text{loc}} \cdot \frac{1 + (d-1)\frac{p_{\text{max}}}{a}}{\left(1 + \frac{p_{\text{max}}}{a}\right)^{d-1}} \right] \quad (3.18)$$

and

$$\Delta E_e^{\text{loc}} = x^{\text{loc}} \cdot \frac{S_e}{2\pi a^2} \cdot (d-1)(d-2) \left(1 + \frac{p}{a} \right)^{-d} \quad (3.19)$$

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. (2.11) and (2.13), 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 [27] with a_{ZBL} the screening

⁴These tables have been obtained by H. Hofsäss, University of Göttingen, by running SRIM-2013 in a double-loop. We thank H. Hofsäss for providing the tables.

length of the ZBL interatomic potential [37].

$$a = f \cdot a_{\text{OR}} (= f \cdot a_{12}/0.3) \quad (3.20)$$

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

$$x^{\text{nl}} \sim E^q \quad (3.21)$$

(models #1, #2, and #5), or

$$x^{\text{nl}} \sim S_e(E)^q \quad (3.22)$$

(all other models) where q is determined by the program from x^{nl} values at two energies. Note that Eq. (2.20) and Eq. (2.19) are equivalent when $S_e \sim E^p$ such as in the Lindhard like model Eq. (2.15). Default values of the parameters are provided by the program for all ions in silicon [?, ?, ?, ?, ?, ?]. 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 [?, ?].

| parameter | IMSIL name | to be specified in record |
|--|------------|---------------------------|
| target density N | DENSITY | &MATERIAL |
| stopping power model | SEMODEL | &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 | SEFILE | &SEPAR |
| nonlocal fraction model | XNLMODEL | &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 [2] 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:

$$\frac{\partial\sigma}{\partial\Omega} = \left[\frac{Z_1 Z_2 e^2}{2E} \right]^2 \cdot \frac{1}{\sin^4 \psi} \cdot \frac{\left[\sqrt{1 - \left(\frac{m_1 \sin \psi}{m_2} \right)^2} + \cos \psi \right]^2}{\sqrt{1 - \left(\frac{m_1 \sin \psi}{m_2} \right)^2}} \quad (3.23)$$

$e^2 = 1.44 \times 10^{-13}$ 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:

$$K = \left[\frac{\sqrt{1 - \left(\frac{m_1 \sin \psi}{m_2} \right)^2} + \frac{m_1 \cos \psi}{m_2}}{1 + \frac{m_1}{m_2}} \right]^2 \quad (3.24)$$

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

$$P = \frac{1}{2\pi x_{\text{rms}}^2} \cdot \exp\left(-\frac{r^2}{2x_{\text{rms}}^2}\right), \quad (3.25)$$

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:

$$P_{\text{RBS}} = \frac{\partial\sigma}{\partial\Omega} \cdot P \cdot N_d \quad (3.26)$$

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

Input and Output Files

In this chapter, the data files which are required or produced by IMSIL are listed, and their format is specified. If IMSIL is run by calling its executable directly, all files are searched for or generated in the current working directory (i.e. from where the executable is called). Normally, IMSIL requires files from the tables directory, so the required tables (or subdirectories of the tables directory) must be copied or linked to the current working directory.

It is recommended to use the Python scripts `imsil.py` and `mimsil.py` as described in Section ?? to start IMSIL. If these scripts are used and the name of the input file is `<xxx>.inp`, a directory `<xxx>` is created and `<xxx>.inp` is copied to `<xxx>/INP`. The same is done with all other files `<xxx>.*`. In addition, the script changes to the new directory `<xxx>`, so it becomes the working directory. In addition, the paths to the tables directory and the input file directory (the directory of `xxx.inp`) are passed to IMSIL. When IMSIL attempts to read a file, it first searches the working directory (`<xxx>`), then the input file directory, and finally the tables directory. As an exception, the `INP` file is only searched for in the work directory (it should always be found). The path to the tables directory is hard-coded in the `imsil.py` script. Output files are always generated in the working directory.

After completion of an IMSIL run, all files `<EXT>` in the working directory `<xxx>` (where `<EXT>` stands for an arbitrary file name) are copied to the input file 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 working directory is changed to `example1`, and IMSIL is run. The simulation generates an output file `OUT` in the working directory, which is then copied to the input file directory as `example1.out`. The same is done with any other files that are created during the run.

If execution of IMSIL is delegated to a remote computer, the script copies files obeying the pattern `<xxx>.*` in the input file directory (where the name of the input file is `<xxx>.inp`) as well as the tables directory with its files to the remote computer. All other files have

to be copied by hand before running IMSIL.

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 | — | — |
| (-) | Geometry file (used if NDIM=2 and GEOMFILE specified) | GEOMFILE | &GEOMETRY |
| (-) | Signed distance function file | SDFFILE | &GEOMETRY |
| (SCATTAB) | Scattering table file (used if QUAD='no' in record &SNPAR) | SCATFILE | &SNPAR |
| (ZBL) | Screening function file (used if QUAD≠'no' and if SCRFUN='scrfile' in record &SNPAR) | SCRFILE | &SNPAR |
| (ZBLspec) | Screening coefficient file (used if QUAD≠'no' and if SCRFUN='coefffile' in record &SNPAR) | COEFFILE | &SNPAR |
| (-) | Random electronic stopping file (used if SEMODEL='table' in record &SEPAR) | SEFILE | &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) | BKFILE | &SETUP |
| (-) | Histogram = histogram profiles necessary to initialize 1-D histogram of desired atom index (used if USEHIS=T in record &SETUP) | HISFILE | &SETUP |
| (-) | Cell file = 1-D cell-based description of the target containing the densities of each species (used if USECELL=T in record &SETUP) | CELLFILE | &SETUP |
| (-) | Trajectory file used to initialize the ions (used if USETRA=T) | TRAFILE | &ION |

The format of the INP file is descibed in detail in Chapter 4. The geometry file is an input file that (usually) contains only &GEOMETRY records.

The signed distance function file must contain the following lines in a header:

```

Dimension: <ndim>
Unit: Angstrom
nX: <nx>
dX: <dx>
nY: <ny>
dY: <dy>
nZ: <nz>
dZ: <dz>

```

The items in angled brackets must be replaced by numerical values. `ndim` denotes the number of dimensions (2 for 2-D, 3 for 3-D). `nx`, `ny`, `nz` denote the number of grid lines and `dx`, `dy`, `dz` the grid spacings in x , y , and z direction, respectively. `dx`, `dy`, and `dz` are either single values, indicating constant grid spacing, or `nx-1`, `ny-1`, and `nz-1` comma-separated numbers, respectively. The header is terminated by the line

Values :

The SDF values are then read by the FORTRAN statements

```
DO IZ = 1, NZ
  DO IY = 1, NY
    READ (22,*) (SDF(IX,IY,IZ,IR), IX=1,NX)
  END DO
END DO
```

Note the region index `IR`. An SDF file must be provided for each region and for the surface of the union of all regions. The latter is assigned to region 0.

The scattering table file `SCATFILE` 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. (??). `PCUTOFF` is the screened cut-off impact parameter, beyond which the deflection angle will be set to zero. `COTTHH` 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 `SCRFILE` 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 Φ the value of the screening function $\Phi(r)$, see Eq. (??). If `SCRFILE='ZBL'`, then the element specific ZBL screening functions provided in the subdirectory `screentab/zbl` of the `tables` directory are used.

The screening coefficient file `COEFFFILE` 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,N)
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. (??). The number N of terms in Eq. (??) can be specified in a header of a file such as

```
# nexp = 4
```

If unspecified, $N=4$ is assumed. If the a_i 's do not sum up to unity, they are scaled so they do.

When the header of the `COEFFFILE` file contains the string `rmax`, an additional value is read from each line of the file representing the maximum interaction distance. For instance,

```
# nexp = 3, rmax
```

specifies that the `COEFFFILE` file contains the parameters for three exponentials and the linear term, compare Eq. (??). They are read by

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

The random electronic stopping power file `SEFILE` 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=1,NE
  READ (122,*) E(IE), DEDX(IE)
ENDDO
```

The file can begin with an arbitrary number of comment lines beginning with the `*` or `#` 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. Below $E(1)$, the electronic stopping is interpolated linearly between zero energy/zero stopping and the first table entry. Above $E(NE)$, electronic stopping is set to zero.

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 BKFILE 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 BKFILE, 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 HISFILE 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 3.3 and 3.6.

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 [21]), 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

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

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

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

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

These moments are reported together with their 1- σ standard deviations (corresponding to the statistical uncertainties), where the standard deviations are give in the lines following each of the moments.

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. (3.1)–(3.4) 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 ??).

4.3 1-D Histogram Files

1-D histogram files store densities as a function of the depth coordinate (z) or other distributions with respect to a spatial coordinate, energy, or angle. Files storing densities are **HIS**, **CELL**, **HISEE**, and **HISNE**. These densities take the implantation dose **DOSE** into account. As an example, the nuclear energy deposition density $f_{\text{NE}}(z)$ 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 \AA^{-2} , and the units of $f_{\text{NE}}(z)$ are $\text{eV}/\text{\AA}^3$. For the **HIS** and **CELL** files, the densities are multiplied by 10^{24} before they are written to the file in order to obtain the more common units cm^{-3} . For the electronic and nuclear energy deposition histograms, it is often desirable to have units of $\text{eV}/\text{\AA}$. This can be achieved by performing the simulation with a dose of $\text{DOSE} = 10^{16} \text{ cm}^{-2} = 1 \text{ \AA}^{-2}$, which is the default in IMSIL if **DOSE** is not specified.

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'). **It is generally not recommended to record 1-D densities when the dose is specified other than as an area dose.** 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. (3.5) for **DOSE** will have units \AA^{-1} . As a result, the units of $f_{\text{NE}}(z)$ will be $\text{eV}/\text{\AA}^2$. If the energy density per \AA^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. For the **HIS** and **CELL** files, the densities are still multiplied with 10^{24} before they are written to the file, so their units will be $\text{\AA}/\text{cm}^3$ in the above example, which certainly is not desirable.

All other distribution do not include the implantation dose **DOSE**. For instance, the angular distribution of backscattered atoms f_{AB} is the probability density function times the backscattering/sputtering yield Y_{B} , so its integral evaluates to the yield

$$\int_{-90^\circ}^{90^\circ} f_{\text{AB}}(\alpha) d\alpha = Y_{\text{B}}. \quad (4.6)$$

Generally, histograms are adjusted by IMSIL to accomodate all scores, either by extending their initially specified size or by coarsening. Exceptions are the **HISD** and **HISIV** files, which are not adjusted.

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 |
| HISSET | Energy distribution of transmitted atoms | LHISSET |
| HISAT | Angular distribution of transmitted atoms (2-D angle) | LHISAT |
| HISAAT | Angular distribution of transmitted atoms (3-D angle) | LHISAAT |
| 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

¹The columns 2 to N1+N2+1 in the histogram files correspond to the atom indices as defined in Section 4.2.

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.

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 [?]. 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 [?]. 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 a certain depth interval (POSIVMIN and POSIVMAX of the OUTPUT record).

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 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 $\text{DOSE} = 10^{16} \text{ cm}^{-2} = 1 \text{ }^{-2}$ 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 $[\text{Å}^{-2}]$).

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 |
| PDV | Vacancy positions after recombination | LCAP LTRA, ITRA | &DAMAGE &OUTPUT |
| PDI | Interstitial positions after recombination | LCAP LTRA, ITRA | &DAMAGE &OUTPUT |
| PMAX | Maximum impact parameters | LPMAXTAB | &SNPAR |
| POK | Interstitial positions before melting | LPOCK LTRA, ITRA | &DAMAGE &OUTPUT |
| SE | Table of stopping powers | LTAB | &SEPAR |
| 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 ??.

The point defect files **PDV** and **PDI** contain the positions of vacancies and interstitials, respectively, after recombination when the capture radius model is activated (**LCAP=T**). In addition, the atom index as defined in Section 4.2 is given for each point defect:

```
DO, I=1,N
  WRITE (34,*) X(I),Y(I),Z(I),ATOM(I)
ENDDO
```

where **N** denotes the number of vacancies and interstitials, respectively, in each cascade. These files are only written if **LCAP=T** and **ITRA=1** and **ITRA=3** are set, respectively.

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 2.4). **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 4.6).

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 amorphous pocket file POK contains the positions of the interstitials and the melting radius before amorphous pocket formation:

```
DO, I=1,N
  WRITE (34,*) X(I),Y(I),Z(I),RMELT(I)
ENDDO
```

where N denotes the number of interstitials in each cascade. This file is only written if the amorphous pocket model is activated (LPOCK=T) and if ITRA=3 is set.

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 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 2.4), 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 ??).

Chapter 5

Input Parameters

The input language of IMSIL uses the FORTRAN90 syntax for NAMELIST input. Each **input record** in the input file **INP** corresponds to a namelist in the program. It contains a set of input parameters. An input record starts with an **&**, followed by the name of the record (= the name of the namelist) 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. Comments are typically placed at the beginning of the file. Comments may also be placed inside a record after an exclamation mark “!”. The comment then extends to the end of the line.

While all names are case-insensitive, in this manual (except for the Examples Chapter), 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 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.

Parameter specifications for a namelist may be distributed across multiple input records:

```
&NAME PARNAME1=value1 ... /  
&NAME PARNAME1=value3 PARNAME2=value2 ... /
```

If input parameters are repeated, the later definitions override the earlier one(s), with the exception of simple arrays of arbitrary size, see below. In the example above, **PARNAME1=value3**.

There are two kinds of array parameters in IMSIL: First, **simple arrays** are one-dimensional arrays of either fixed size (often the size is 2 or 3), or their size is determined by the number of elements given (“arbitrary” size). Simple arrays 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 ... /
```

When more than one input record is specified for the same namelist and specifications for the same simple array of arbitrary size are given, unless otherwise noted, later array values are appended to earlier ones. Thus,

```
&NAME ARRAYNAME=value1,value2 ... /
&NAME ARRAYNAME=value3 ... /
```

defines the array (value1,value2,value3) like in the example above, if **ARRAYNAME** is a simple array of arbitrary size. If **ARRAYNAME** is a simple array of fixed size, then **ARRAYNAME=value3**, i.e., only the first element of **ARRAYNAME** is given a value. The maximum number of array values that can be specified on each input record is 20.

Second, **index variable arrays** are arrays whose values may be specified for individual indices using index variables. For instance,

```
&MATERIAL REGION=2 NAME='SI3N4' ... /
&MATERIAL REGION=1 NAME='SI02' ... /
&MATERIAL REGION=3 NAME='SI' ... /
```

specifies the array **NAME(1:3)=('SI02','SI3N4','SI')**, where in each record the index variable **REGION** indicates the index of the array element whose value is to be set.

If an index variable array appears on a record that does not contain the index variable(s), the array values for all unspecified indices are set to the specified value. On the other hand, on records containing index variables, no parameters must be specified that are not arrays of that index.

Index variable arrays are used for arrays depending on some standard indices like the region or atom index, and for multi-dimensional arrays.

As a convenient feature of NAMELIST input, only those parameters need to be specified whose values deviate from the predefined default values (given by the program before reading the NAMELIST record). A few parameters are obligatory, IMSIL will require them to be specified. Examples of actual input files are given in Chapter 5.

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 **HISFILE** 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 **HISFILE**. Relevant only if **USEHIS=T**.

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

BKFILE — 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

CELLFILE — 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 3). Relevant only if USECELL=T.

Type: character string
Default: – (obligatory if USECELL=T)
Range: any name (≤ 80 characters) of an IMSIL cell file

CELLREC — Record to be read from cell file

This variable specifies the index of the record to be read from the CELL or CELL2 file, where record means the data describing one snapshot of the cells. CELLREC=0 means the last record will be read.

Type: integer
Default: 0
Range: $\text{CELLREC} \geq 0$

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

HISFILE — 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 3). 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: 15678
 Range: $1 \leq \text{IARAND} \leq 70000$ for `RNG='Haas'`
 $1 \leq \text{IARAND} \leq 131071$ otherwise

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: 67638
 Range: $1 \leq \text{IBRAND} \leq 90000$ for `RNG='Haas'`
 $1 \leq \text{IBRAND} \leq 262143$ otherwise

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: 5421
 Range: $1 \leq \text{IRAND} \leq 10000$ for `RNG='Haas'`
 $1 \leq \text{IRAND} \leq 16383$ otherwise

LDAMDYN — Dynamic damage consideration flag

This variable specifies whether damage accumulated during the simulation is considered for recombination with new damage and/or for selecting the collision partners in a crystalline region.

Type: logical
 Default: LDAM of the `&DAMAGE` record
 Range: T, F

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 3) 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 2.4). 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: 6103

Range: $1 \leq \text{MRAND} \leq 8000$ for RNG='Haas'
 $1 \leq \text{MRAND} \leq 16383$ otherwise

NATOM — Number of atom species

This variable specifies the number of atom species contained in the ion and the target regions. E.g. for BF₂ implantation in a two-layer structure of SiO₂ 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 BF₂) 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). Alternatively, for 2-D and 3-D targets (NDIM=2 or NDIM=3) the geometry may be specified by signed distance functions. See the &GEOMETRY record

Type: integer

Default: 1

Range: 1, 2, 3

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 read from the backup file. If NION is not a multiple of NTHREAD, it is adjusted to the next higher multiple of NTHREAD.

Type: integer

Default: – (obligatory)

Range: > 0

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). If NIONBK is not a multiple of NIONCHUNK, it is adjusted to the next higher multiple of NIONCHUNK. Note that only updated histograms are stored in the simulation. Therefore, temporary ion and damage distributions are lost if NIONHIS>0 and NIONBK is not a multiple of NIONHIS.

Type: integer

Default: 0

Range: ≥ 0

NIONCHUNK — Number of simulated ions per chunk

This parameter specifies the number of ions to be simulated in one “chunk”. Larger chunks slightly reduce the simulation time, while smaller chunks allow more refined damage and/or geometry updates and/or output on the backup file (see NIONHIS, NIONGEOM, and NIONBK), since these actions can only be done after integer multiples of NIONCHUNK simulated ions, and at the end of the simulation. If NIONCHUNK is not a multiple of NTHREAD, it is adjusted to the next higher multiple of NTHREAD. Usually, the default value of NIONCHUNK can be used, therefore it often is not necessary to specify this parameter in the input file.

Type: integer

Default: NIONHIS, if LDAMDYN=T or LDYN=T, and NIONHIS>0

NIONGEOM otherwise, if LDYN=T

NION otherwise

greatest common factor of value above and NIONTRA, if ITRA or

LTRA specified on &OUTPUT record

Range: > 0

NIONGEOM — Number of ions for geometry updates

This parameter specifies the number of implanted ions after which the cells are checked for necessary geometry updates in a dynamic simulation (LDYN=T). See also **FRACUPD** of the **&GEOMETRY** record. Note that updating the geometry requires some computational effort in 1D, and considerable computational effort in 2D. This effort decreases with larger **NIONGEOM** at the expense of accuracy. On the other hand, too infrequent geometry updates affect the simulation results if LDYN=T. If **NIONGEOM** is not a multiple of **NIONHIS**, it is adjusted to the next higher multiple of **NIONHIS**. **NIONGEOM** is ignored if LDYN=F.

Type: integer
 Default: **NIONHIS**, if **NIONHIS** is defined and > 0
 NIONCHUNK otherwise, if **NIONCHUNK** is defined
 NTHREAD otherwise
 Range: > 0

NIONHIS — Number of ions for histogram updates

If greater than zero, this parameter specifies the number of implanted ions after which the ion and damage histograms (LDAMDYN=T) or the cell densities (LDYN=F) are updated. All other histograms (e.g., the histograms of backscattered and transmitted atoms) are not affected by **NIONHIS**; they are only updated at the end of the simulation. **NIONHIS=0** means that the histograms or cell densities are updated whenever an event leads to a contribution to a histogram or cell density during the development of a collision cascade. This is only allowed when **NTHREAD=1**. Note that updating histograms requires some additional computational effort, in particular in parallel processing. This effort decreases with larger **NIONHIS** at the expense of accuracy. On the other hand, too infrequent damage updates affect the simulation results if LDAMDYN=T. If **NIONHIS** is not a multiple of **NIONCHUNK**, it is adjusted to the next higher multiple of **NIONCHUNK**. **NIONHIS** is ignored if LDAMDYN=F and LDYN=F.

Type: integer
 Default: 0, if **NTHREAD** = 1
 NIONCHUNK, if **NTHREAD** > 1 and **NIONCHUNK** is defined
 NTHREAD, if **NTHREAD** > 1 and **NIONCHUNK** is not defined
 Range: $0 \leq \text{NIONHIS} \leq \text{NION}$, if **NTHREAD** = 1
 $1 \leq \text{NIONHIS} \leq \text{NION}$, if **NTHREAD** > 1

NIONOUT — Number of ions for output

This parameter specifies the number of ions after which output is written to the **CELL**, **CELL2**, and **SURF** file in case of a dynamic simulation (**LDYN=T**). **NIONOUT** is rounded to the next lowest integer multiple of **NIONGEOM** (after any adjustments to **NIONGEOM**). In addition, $\text{NIONOUT} \geq \text{NIONGEOM}$ is enforced. If **NION** is not an integer multiple of **NIONOUT**, an additional data record is written at the end of the simulation.

Type: integer
 Default: **NIONGEOM**
 Range: > 0

NIONTRA — Number of ions for trajectory output

This parameter specifies that trajectories shall be written to the **TRA** file for every **NIONTRA**th ion. **NIONTRA=1** means that trajectory output shall be written for every ion. If **NIONTRA** is not a multiple of **NIONCHUNK**, it is adjusted to the next higher multiple of **NIONCHUNK**. This parameter can be used for reducing the amount of output to the **TRA** file. **NIONTRA** is ignored unless **ITRA** or **LTRA** of the **&OUTPUT** record is set.

Type: integer
 Default: 1
 Range: > 0

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: $\text{NR} > 0$

NTHREAD — Number of threads

This variable specifies the number of threads. It overwrites the number of threads specified as an environment variable or during compilation. It is ignored when the code is compiled without multiprocessing or in the coarray version of the code.

Type: integer
 Default: as specified by an environment variable or during compilation
 Range: $\text{NTHREAD} > 0$

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** \geq 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. It is therefore not possible to report **MRAND**, **IARAND**, **IBRAND**, and **IRAND** at the end of the simulation. However, different seeds lead to different random number sequences of Knuth's random number generator.

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 **BKFILE** 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 `CELLFILE` parameter. If `USECELL=T`, the cell file specified by `CELLFILE` 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 3). 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 `HISFILE` 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 of regions 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 can be specified either by a closed polygon or a signed distance function (SDF). 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 can be specified as region 0. If it is not specified, it is determined automatically from the other regions. There must not be undefined areas within the circumference, and no regions may overlap.

Alternatively, the geometry of each region as well as of the union of all regions can be defined by SDFs. SDFs must be calculated externally, and are provided as files, see the parameter **SDDFILE**.

In both cases of 2-D geometry definition, **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. In case of geometry definition by SDFs, the internal mesh coincides with the mesh of the SDFs.

3-D geometry definition is possible via SDFs only.

It is possible to impose cylindrical (**LCYLX=T**, **LCYLY=T**, or **LCYLZ=T**) or spherical (**LSPHER=T**) symmetry. Specification of cylindrical geometry requires 1-D or 2-D geometry definition, where the distance along the axis of symmetry replaces the x coordinate, while the distance from the axis replaces the z coordinate. Specification of spherical symmetry can only be combined with 1-D geometry definition, where 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

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

GEOMFILE — 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 before the first `&GEOMETRY` record of the input file. As a consequence, any specifications in the input file will overwrite corresponding specifications in the geometry 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. `GEOMFILE` is ignored for 1-D geometries (`NDIM=1`).

Type: character string

Default: –

Range: Any valid file name (≤ 80 characters)

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 and NDIM=1 or NDIM=2.

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 and NDIM=1 or NDIM=2.

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** and **NDIM=1** or **NDIM=2**.

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**. **NBOX2** must not be specified together with **SDDFILE**.

Type: simple array (2) of integer
 Default: 1000, **NBOX2(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. Specification of **POINTS** requires the index variable **REGION** on the same record. A maximum of 20 points may be specified on one record. Specifications of **POINTS** on different records for the same **REGION** are joined together. In this way, larger arrays of points may be split to several records.

Type: index variable array (**REGION**) of arbitrary-size simple arrays of integer
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. A maximum of 20 positions may be specified on one record. Specifications of **POSIF** on different records are joined together. In this way, larger arrays of positions may be split to several records. The units of **POSIF** are Å.

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

SDFFILE — SDF file

This parameter specifies the name of a signed distance function file. An SDF file must be specified either for all regions or for none. For 2-D simulations (**NDIM=2**), the geometry must be specified either by SDF files or by **POS** and **POINTS**.

Type: index variable array (**REGION**) of character strings
 Default: –
 Range: any name (≤ 80 characters) of a file containing SDF data.

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. WBOX2 must not be specified together with SDDFILE.

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 boundary of the roughness layer). Note that specifying a WROUGH > 0 effectively removes material from the target and moves the “average” surface WROUGH/2 into the target. 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: $XPER(2) > XPER(1)$

XSCALE — Scaling factor in x-direction

This parameter specifies a scaling factor for the x coordinates of the polygon points as specified by POS. All values POS(1) will be multiplied by XSCALE. Note that XPER and WBOX2(1) will not be scaled.

Type: real
Default: 1
Range: arbitrary

ZSCALE — Scaling factor in z-direction

This parameter specifies a scaling factor for the z coordinates of the polygon points as specified by POS. All values POS(2) will be multiplied by ZSCALE. Note that WBOX2(2) will not be scaled.

Type: real
Default: 1
Range: arbitrary

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 3.3,

```
&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 3.3). 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. NOTE: Due to a limitation of the code, ion atom masses specified on **&ATOMS** are currently ignored. Use **MASS** on the **&IONS** record instead to define ion atom masses.

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 ??.

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

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 $\text{DIVERG}/\sqrt{2}$ 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 BF₂ implantation, etc. The number of atom species (2 in the case of BF₂), 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 string

Default: — (obligatory)

Range: any chemical name of a molecule (≤ 80 characters)

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**, **LCYLX=LCYLY=LCYLZ=LSPHER=F**).

Type: logical

Default: **NDIM>1.OR.LCYLX.OR.LCYLY.OR.LCYLZ.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. **TILT** is an alias of **TILTY**. Only one of **TILT** and **TILTY** must be specified.

Type: real

Default: 0

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

TILTX — Tilt angle about x axis

This parameter specifies the tilt angle about the x axis. The units of **TILTX** are degrees.

Type: real

Default: 0

Range: $0 \leq \text{TILTX} < 90$

TILTY — Tilt angle about y axis

This parameter specifies the tilt angle about the y axis. The units of **TILTX** are degrees. Only one of **TILT** and **TILTY** must be specified.

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

TRAFILE — 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 3). 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

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 **TRAFILE** 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 3 for details).

Type: character string
 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 4.5). The geometry of the target regions is specified in the **&GEOMETRY** record (see Section 4.9).

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 4.2.

DENSITY — Atom density

This parameter specifies the number of target atoms (irrespective of their species) per cm^3 . E.g., **DENSITY=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 strings
 Default: – (obligatory)
 Range: any chemical name of a material (≤ 80 characters)

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 the **&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 strings

Default: 'no'

Range: any character string (≤ 80 characters)

5.7 &DAMAGE

This record is used to describe damage related parameters. For a description of many of the parameters see Section 2.3. 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 4.2.

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 $\text{DAMAMO} < \text{DAMSAT}$. **DAMSAT** is used to calculate the default value of **RCAP**, if **LCAP=T**. 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 2.4). 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

DPARELAX — Critical number of displacements per target atom for cell relaxation

This parameter specifies the minimum number of displacements per target atom between cell relaxations which is required so that any of the cell corners may be moved during relaxation. In converting damage energy into displacements per atom the modified Kinchin-Pease model is used. The units of DPARELAX are 1. Note that the energy density accumulated depends on the implanted dose between relaxations. This dose equals DOSE*NIONUPD/NION with DOSE and NION from the IONS record and NIONUPD from the GEOM record. DPARELAX is only effective for RELAXMODEL=1.

Type: real
Default: 10^{-3}
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 $\text{FPOCK} = (4\pi\text{DENS}(\text{IR})/3)^{1/3}\Delta r$, 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: $0 \leq \text{FRACRG} \leq 1$

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: $0.0264 \times M_{\text{Ion}}$
 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 PMAX 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. LAMZON is only relevant for the selection of collision partners in crystalline regions if LDAM=T and NDAMDIM=1.

Type: logical
Default: F
Range: T if LDAMDYN=T
T, F otherwise

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. LCAP=T must not be set if LPOCK=T. LCAP is ignored unless LDAM=T and LRcoil=T.

Type: logical
Default: F
Range: T, F

LDAM — Refined damage model flag

This parameter specifies whether a refined damage model is used for the calculation of interstitial and vacancy distributions. For **LDAM=F**, either damage is not recorded at all (**LRCOIL=F**), or exactly one interstitial is generated when and where a projectile stops and one vacancy is generated when and where a target atom is recoiled (**LRCOIL=T**). For **LDAM=T**, damage recombination or multiplication is treated depending on other flags: If **LCAP=F**, **LPOCK=F**, the model including the damage recombination factor **FREC**, **DAMSAT**, and **DAMAMO** as described in Section 2.3 is used. When **LCAP=T**, the capture radius model is invoked. When **LPOCK=T**, the amorphous pocket model is used.

The value of **LDAM** is also the default value of **LDAMDYN** of the **&SETUP** record. **LDAMDYN** determines whether damage is taken into account in the selection of collision partners in crystalline regions. The full models invoked by **LDAM=T** and the usage of damage for collision partner selection require the damage distributions to be kept up-to-date during the simulation. The frequency of damage updates is specified by **NIONUPD**.

Note that moments and histograms of the damage distributions are only output if the appropriate flags are set on the **&OUTPUT** record. Vacancy distributions are only recorded if **LRCOIL=T**. 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). **LIBIIA=T** requires **LDAM=T** and **NDAMDIM=1**. The amorphous zones as modified by the IBIIA effect are currently only reported to the OUT file. It is not considered for the selection of collision partners.

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. LPOCK=T must not be set if LCAP=T. LPOCK is ignored unless LDAM=T and LRCOIL=T.

Type: logical
Default: F
Range: T, F

LRCOIL — Recoil cascade flag

This parameter specifies whether the recoil trajectories are to be simulated. LRCOIL=T is required for sputter yields to be calculated and is prerequisite for some damage models (LCAP=T, LPOCK=T). LRCOIL also determines whether vacancy distributions are recorded separately from interstitial distributions. For LRCOIL=F or ELIMKP > ED, energy deposition is calculated using the modified Kinchin-Pease model. Damage is either not recorded (LDAM=F, or is calculated with a probabilistic version of the model including the damage recombination factor FREC, DAMSAT, and DAMAMO as described in Section 2.3. See also LDAM.

Type: logical
Default: F
Range: T, F

LRCOIL1 — Recoil cascade before reentering flag

This parameter specifies whether projectiles with the reentered flag unset can generate recoils whose trajectories are followed. The reentered flag is set if a projectile or one of its parents has been emitted from a surface and has reentered the target at a different point in space. The reentered flag is always unset in a cartesian 1D simulation. Note that for recoils to be followed at all, also LRCOIL=T is required. The purpose of LRCOIL1 is to provide the possibility of switching off recoil generation before reentering while keeping recoil generation after by setting LRCOIL=T, LRCOIL1=F, LRCOIL2=T.

Type: logical
Default: T
Range: T, F

LRCOIL2 — Recoil cascade after reentering flag

This parameter specifies whether projectiles with the reentered flag set can generate recoils whose trajectories are followed. The reentered flag is set if a projectile or one of its parents has been emitted from a surface and has reentered the target at a different point in space. The reentered flag is always unset in a cartesian 1D simulation. Note that for recoils to be followed at all, also `LRCOIL=T` is required. The purpose of `LRCOIL2` is to provide the possibility of switching off recoil generation after reentering while keeping recoil generation before by setting `LRCOIL=T`, `LRCOIL1=T`, `LRCOIL2=F`.

Type: logical
 Default: T
 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

MODBULK — Bulk modulus

This parameter specifies the bulk modulus used in the viscoelastic relaxation model. The units of `MODBULK` are Pa. Ignored unless `RELAXMODEL='viscoelastic'`.

Type: real
 Default: 4.8×10^{10}
 Range: > 0

MODSHEAR — Shear modulus

This parameter specifies the shear modulus used in the viscoelastic relaxation model. The units of `MODSHEAR` are Pa. Ignored unless `RELAXMODEL='viscoelastic'`.

Type: real
 Default: 3.3×10^{10}
 Range: > 0

NDAMDIM — 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, or for advanced damage calculation (LDAM=T. Note that not all dimensionalities are possible depending on the choices of DOSEUNITS and XINIT and YINIT of the &IONS record (see Section 3). 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 NDAMDIM.

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: $1 \leq \text{NDAMDIM} \leq 3$ 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`, `LCYLX=LCYLY=LCYLZ=LSPHER=F`).

Type: logical
 Default: `NDIM>1.OR.LCYLX.OR.LCYLY.OR.LCYLZ.OR.LSPHER`
 Range: T, F

RELAXMODEL — Relaxation model

This parameter specifies the model used for target relaxation in a dynamic simulation. `RELAXMODEL='full'` specifies full relaxation of the target densities along a steepest descent path of the square of the deviation of cell volumes from the ideal values. `RELAXMODEL='viscoelastic'` specifies a viscoelastic model with radiation induced viscosity.

Type: character(80)
 Default: 'none'
 Range: 'none', 'full', 'viscoelastic'

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

VISCOSITY — Radiation induced viscosity

This parameter specifies the radiation induced viscosity used in the viscoelastic relaxation model. The units of **VISCOSITY** are Pa dpa. The number of displacements per atom (dpa) is calculated from the damage energy using the modified Kinchin-Pease model. Ignored unless **RELAXMODEL**='viscoelastic'.

Type: real
Default: 0.966×10^9
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 2.1 and 2.2. 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 4.2.

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

COEFFILE — 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**='coeffile'. Files are assumed to be located in the subdirectory '**screencoefs**' of the search path.

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

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: $0 < \text{EF} \leq \text{ENERGY}$

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**. If larger than the size of the simulation domain, **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**). In 2-D, the size of the simulation domain is taken as the diameter of the rectangle circumscribing the simulation domain. The units of **FFPMAX** are \AA .

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

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 `SCATFILE`. Either both or none of `QUAD` and `QUADTIME` must be `'no'`.

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

SCATFILE — 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 SCATFILE. Files are assumed to be located in the subdirectory 'scattab' of the search path.

Type: index variable array (ATOM2,ATOM1) of character strings
 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 (≤ 80 characters) of a user-supplied file containing a scattering table

SCRFILE — 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 COEFFFILE is either unspecified or overridden by specification of SCRFILE. COEFFFILE and SCRFILE must not appear on the same input record. Specification of COEFFFILE or SCRFILE on later input records override those on earlier ones. SCRFILE='ZBL' and SCRFILE='DMol' are interpreted as SCRFILE='zbl/XXYY' and SCRFILE='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 strings
 Default: 'ZBL'
 Range: 'ZBL', 'DMol', 'zbl/XXYY', 'dmol/XXYY' (for the meaning of XXYY see above), or the name (≤ 80 characters) 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'), 'coefffile' as a reference to a coefficients file, or 'scrfile' as a reference to a screening function file. In the two latter cases, the file is specified by COEFFFILE and SCRFILE, respectively. SCRFUN is ignored if QUAD='no'. ZBL refers to the universal ZBL potential. SCRFUN is not used for the calculation of the time integral, the universal ZBL screening function is used for this purpose.

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

Default: 'ZBL'

Range: 'ZBL', 'KrC', 'Moliere', 'coefffile', 'scrfile'

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 strings

Default: 'ZBL' for SCRFUN='ZBL'

'Firsov' for SCRFUN='KrC' or 'Moliere'

if specified in the coefficients file, according to this specification,

otherwise 'ZBL', for SCRFUN='coefffile'

'1' for SCRFUN='scrfile'

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 2.5. 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 4.2.

CORLIN — Correction to Lindhard stopping

This parameter specifies the correction factor k_{corr} in the modified Lindhard stopping formula $S_e = k_{\text{corr}} k_L \sqrt{E}$. **CORLIN** is only effective if **SEMODEL**='Lindhard'.

Type: index variable array (**ATOM2**,**ATOM1**) of real
 Default: see [14] 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 **SEMODEL**='Lindhard'.

Type: index variable array (**ATOM2**,**ATOM1**) of real
 Default: 1.43 for **Z1**=1 [12]
 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 **SEMODEL**='Lindhard'.

Type: index variable array (**ATOM2**,**ATOM1**) of real
 Default: $40 * Z1 * Z2$ for **Z1**=1 [12]
 $100 * Z1 * Z2$ 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. It should not be used with **XNLMODEL**='E'. If it is, it is allowed only as the last **ENL** value, and x^{nl} at the stopping power maximum will not equal the last **XNL** value. Instead, x^{nl} at the last but one **ENL** value will behave similarly as for **XNLMODEL**='Se'. If **ENL** values for the same atom indices appear on different records, later specifications override earlier ones.

Type: index variable array (**ATOM2**,**ATOM1**) of arbitrary-size simple arrays of real
 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 $x^{\text{nl}} = 1$

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 $x^{\text{nl}} = 0$ and **FAC2SCR** the value of f at $x^{\text{nl}} = 1$.

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

LAPSIS — Apsis flag

This parameter specifies whether the apsis (distance of closest approach) of the collision (**LAPSIS=T**) or the impact parameter (**LAPSIS=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**

LOCMODEL — Local electronic stopping model

This parameter specifies the model for the local electronic stopping power. It can be either **LOCMODEL='OR'** for a modified Oen-Robinson model, **LOCMODEL='Firsov'** for a modified Firsov model, or **LOCMODEL='none'** for pure nonlocal electronic stopping.

Type: index variable array (ATOM2,ATOM1) of character strings
 Default: **'OR'**
 Range: **'OR'**, **'Firsov'**, **'none'**.

LTAB — Stopping table flag

This parameter specifies whether tables of the nonlocal and local electronic stopping powers 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.

Type: logical
 Default: F
 Range: T, F

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. **POWFIRS** is only used with **LOCMODEL='Firsov'**.

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 modified Lindhard Bethe-Bloch stopping powers. **POWINT** is only effective if **SEMODEL='Lindhard'**.

Type: index variable array (**ATOM2,ATOM1**) of real
 Default: according to [14] 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 **SEMODEL='Lindhard'**. 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 writing the third column of the stopping power table on the file **SE** if the flag **LTAB** is set.

Type: real
 Default: 0
 Range: ≥ 0

SEFILE — Random electronic stopping file

This parameter specifies the name of the random electronic stopping file. **SEFILE** is only effective if **SEMODEL**='table'.

Type: index variable array (**ATOM2**,**ATOM1**) of character strings
 Default: – (obligatory if **SEMODEL**='table')
 Range: any name (≤ 80 characters) of a file containing random electronic stopping data.

SEMODEL — Electronic stopping model

This parameter specifies the electronic stopping power model. **SEMODEL**='Lindhard' selects the combined modified Lindhard and Bethe-Bloch model. **SEMODEL**='SRIM' selects the SRIM-2013 model. For **SEMODEL**='table' the electronic stopping data ($-dE/dx$ [eV/Å]) will be read from the file **SEFILE**. The options **SEMODEL**='ZBL' for the ZBL85 stopping power and **SEMODEL**='KKK' for the Konac-Klatt-Kalbitzer model are generally not recommended. **SEMODEL**='KKK' is only allowed for **ATOM2** corresponding to Si atoms. **SEMODEL**='none' can be used to switch off electronic stopping.

Type: index variable array (**ATOM2**,**ATOM1**) of character strings
 Default: 'Lindhard' if there is a default value for **CORLIN**
 'SRIM' otherwise
 Range: 'Lindhard', 'SRIM', 'table', 'ZBL', 'KKK', 'none'

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 strings
 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. If XNL values for the same atom indices appear on different records, later specifications override earlier ones.

Type: index variable array (ATOM2, ATOM1) of arbitrary-size simple arrays of real
 Default: 0.15, 0.32 at 1eV, 1keV for Z1=5,
 0.25, 0.50 at 1eV, 1MeV for Z1=15,
 according to [14] for ions specified there,
 0.5 otherwise
 Range: ≥ 0

XNLMODEL — Nonlocal fraction model

This parameter specifies the model for the energy dependence of the nonlocal fraction x^{nl} (see parameter XNL). For XNLMODEL='E', the nonlocal fraction is proportional to a power of the projectile energy E (Eq. 2.20), limited to $x^{nl} \leq 1$. For XNLMODEL='Se', the nonlocal fraction is proportional to a power of the electronic stopping power (Eq. 2.19). XNLMODEL='E' is slightly more efficient than XNLMODEL='Se'

Type: index variable array (ATOM2, ATOM1) of integer
 Default: 'E' for $E_0 < 10 \text{ keV} \cdot (\text{ion atom mass})$,
 'Se' otherwise
 Range: 'E', 'Se'

n

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 \leq 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, see the description of the TRA file in Chapter 3), corresponding to a value of ITRA will be written. Trajectories are only written from thread 1, so it is recommended to use a single thread only when requesting trajectory output. In addition, the PDV and PDI files will be written if LCAP=T (see &DAMAGE record) and an ITRA=1 and ITRA=3, respectively. The POK file will be written if LPOCK=T (see &DAMAGE record) and an ITRA=3.

Type: arbitrary-size 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 (XCYL,YCYL).

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. LHIS=T is required if LDAM=T and NDAMDIM=1.

Type: logical
Default: **T** if LDAM=T and NDAMDIM=1
 F otherwise
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. LHIS2=T is required if LDAM=T and NDAMDIM=2.

Type: logical
Default: **T** if LDAM=T and NDAMDIM=2
 F otherwise
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. LHIS3=T is required if LDAM=T and NDAMDIM=3.

Type: logical
Default: T if LDAM=T and NDAMDIM=3
F otherwise
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**

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). The positions of point defects considered may be restricted to certain depth intervals, see the POSIVMIN and POSIVMAX parameters.

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 3.3 for an explanation of the first redistributive moment.

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

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 3.3 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 LTRA=T and ITRA is not specified, all possible output is written. Otherwise trajectory output is determined by ITRA. Regarding the TRA file and its format see Chapter 3). Trajectories are only written from thread 1, so it is recommended to use a single thread only when requesting trajectory output.

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 $2 \times \text{NBOXA}$. 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

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: 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: 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 accomodate all scores in **NBOX** boxes. The units are Å.

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 accomodate all scores in **NBOX2** boxes. 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 accomodate all scores in `NBOX3` boxes. 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 accomodate 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 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 IMSIL 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> -i 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. `-i` is one of several possible flags and causes IMSIL to run interactively in the foreground. (The default is to run IMSIL in the background, so you can continue to use your command window.)

You can also start several simulations at once:

```
<path-to-imsil.py> -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.

To call the parallel version of IMSIL, enter

```
<path-to-imsil.py> -t 2 example.inp
```

In this case, the parallel version with 2 threads is called. If used together with the `-p` flag, the total number of simultaneous threads is the product of the arguments of the `-t` and `-p` flags. E.g., `-t 8 -p 4` implies 32 simultaneous threads. `-t 0`, which is the default, invokes the sequential version of IMSIL.

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> -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. In addition, there are some requirements on installed libraries on the remote host, which vary between zero (serial version of IMSIL, same version of same operating system) to significant (OpenMP version, different operating system).

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> 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 3.3). |
| <code>read_his2</code> | Read a 2-D histogram file (<code>.his2</code> and other files described in Section 3.4). |
| <code>read_his3</code> | Read a 3-D histogram file (<code>.his3</code> and other files described in Section 3.5). |
| <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 nion=10000 /
&ions       name='As' energy=10 dose=1e13 tilt=7 rotate=15 diverg=0.3 /
&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 4, 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 nion=10000 /
```

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). 10000 projectiles are requested to be simulated. The next line

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

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`. 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 $z = 100000 \text{ \AA}$ ($= 10 \text{ }\mu\text{m}$). 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.

The **ldam** flag is set in the **damage** record (it would also be set by default even if it were not specified in the input file). Therefore, by default, damage is dynamically considered during the simulation for calculating damage increments and for the selection of collision partners in the crystalline region (**ldamdyn=t**). If the program is run with a single thread, the damage histograms are continuously updated during the simulation (**nionhis=0**). This means, as an interstitial or vacancy is generated, the associated damage is calculated and the damage distribution is updated, so it can be used immediately from the next collision on. If the program is run with multiple threads (**nthread** threads), damage distributions are collected in individual histograms per thread, and merged with the overall damage histogram every **nionhis** collision cascades (**nionhis/nthread** collision cascades per thread). By default, this update occurs after every collision cascade per thread. When 1-D histograms of moderate size are used, as in this example, this is a good choice, since the computational effort for joining the histograms is low.

If the **imsil.py** 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 nion=100000 /
&ions       name='B' energy=25 dose=1e15 tilt=7 diverg=0
            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=1 points=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20 /
&geometry   region=1 points=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.

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`.

As in the previous example (Section ??), the damage histograms are updated every `nthread` ions if the program is run with `nthread > 1` threads. Since now the computational cost for merging the 2-D histograms (`ndamdim` of the `&DAMAGE` record is set to `ndim` by default) is significant, the efficiency of the simulation can be improved by specifying, e.g., `nionhis=80` on the `setup` record. Note that increasing `nionhis` too much affects the accuracy of the simulation.

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 nion=50000 nionhis=800 /
&ions       name='B' energy=30 dose=5.E13 tilt=7. rotate=15. diverg=0.3 /
&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=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 /

```

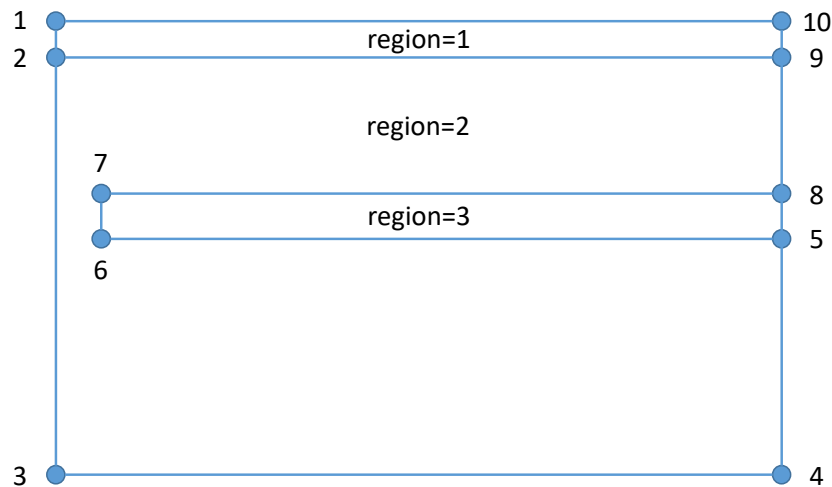


Figure 7.1: Geometry of the multilayer example. Lengths are not to scale. The point and region indices are indicated.

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. ?? . 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 nion=1000000 nionhis=800 /
&ions       name='B' energy=2 dose=1 doseunits='1'
            tilt=7 diverg=0.3 /
&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 ndamdim=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). `ndamdim=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 `ndamdim=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 `wbox2` 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 nion=100000 /
&ions       name='Ar' energy=10 dose=1 tilt=60 diverg=0 /
&material   region=1 name='Si' xtal='no' /
&snpar      chkffpcyl=t /
&damage     ldam=f 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.

Since accurate damage distributions are not of interest in this case, `ldam=f` is set on the `&DAMAGE` record. As a consequence, damage distributions are calculated individually by each thread in a multi-threaded run and merged only at the end of the simulation. This does not affect the results, as the implantation dose of 1 cm^{-2} is (unrealistically) low and damage remains weak throughout the simulation. As a benefit, since the merging of the histograms is done only at the end of the simulation, the computational overhead of the merge operation is largely avoided.

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 runs 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
            nion=#nion#, nionhis=100 niongeom=8
            usecell=#usecell# cellfile='#cellfile#' /
&ions       name='Kr' energy=2 dose=1e15 tilt=60 diverg=0 /
&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 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. The geometry is checked for updates every `niongeom=8` ions and histogram (cell content) updates are requested every `nionhis=100` ions. Since 100 is not an integer multiple of 8, it is increased to 104 by IMSIL internally. 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 `cellfile` 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 `cellfile` 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. `fracupd=0.` specifies that at every geometry update attempt (i.e. after every `niongeom` ions) relaxation of the target densities is performed. If a value larger than 0 were specified, the updates

would be done 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]['cellfile'] = ''
    param[fname]['nion'] = '1000'

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

    fname = 'krdyn_10e15'
    param[fname] = {}
    param[fname]['usecell'] = 't'
    param[fname]['cellfile'] = '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 nion=10000
              atom1=2 atom2=2 ndim=1 usehis=t hisfile='dam2percent.his' /
&ions       name='He' energy=3000 dose=1e14 tilt=0.0 /
&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` `hisfile='dam2percent.his'`). The `.his` file must conform to the format of a 1-D histogram file (see Section 3.3), 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 `hisfile` 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 3.3. The energy box widths are 6000Å. The backscattering angle is specified as 170° (`tilta=10.`).