IMINTDYN81 (Version 8.1)

Ion Matter Interaction - Dynamic

A versatile and fast Monte Carlo Binary Collision Approximation simulation software

Users manual

(first draft)

Prof. Dr. Hans Hofsäss

2nd Institute of Physics

University Göttingen

Germany

Users manual IMINTDYN81 (Version 8.1)

1. Table of Contents

2. Overview

IMINTDYN is a versatile Software for simulation of ion-solid interactions based on the binary collision approximation (BCA) and the Monte Carlo concept [1,2,3]. The program only uses amorphous target structures, but can simulate complex layered target structures. It was developed based on the BCA Monte Carlo software SDTrimSP [4]. The Monte Carlo concept uses weighted random numbers to select the collision partners, the impact parameter, the azimuthal scattering angle, angular or energy distributions of the incoming projectiles etc. Most of the physics used in these programs is described in the book "Computer simulation of ion solid interactions" by Wolfgang Eckstein (Springer, Berlin, 1991) [5]. The program concept of IMINTDYN is similar to SDTRIMSP and is documented in the user's manual of SDTRIMSP Version 6 [4].

IMINTDYN Version 8 typically uses Intel Fortran Compilers, either the older Intel® FORTRAN Compiler, or the INTEL oneAPI compiler [6] (API = application programming interface). The compilation script file automatically selects the proper compiler options based on the installed compiler.

IMINTDYN does not require to be executed on larger computer clusters. The program runs perfectly and efficient on desktop PCs and uses the number of processor cores available. Preferably one should use an up-to date processor such as an intel i9 or an AMD Ryzen Threadripper with typically 24 cores and 48 threads.

The documentation of SDTrimSP V6.00 contains some data of parallel efficiency. These data are obtained with rather old computer clusters such as IBM Regatta 1.3 GHz and LINUX/INTEL clusters dating back to years 2001-2010. Although only demonstrating the efficiency increase with parallel processing, the data give the wrong impression that SDTrimSP should be executed on larger computer clusters.

1.1 Dynamic simulations

Both programs can do simulations in the so-called dynamic mode. In this mode, the elemental composition, structure growth or recession of a target is continuously updated after a small number on incident ions. This update is done after every bunch of default 12 ions. A larger number of projectiles per history can lead to large incremental stoichiometry changes.

To keep incremental stoichiometry changes small, we should use a sufficiently small number of projectiles per history and a sufficiently large number of histories.

A simulation is characterized by three parameters:

1. The total number of projectiles
2. The number of projectiles per history. After every history the target composition is updated in a dynamic simulation. This number should be small enough to keep the incremental stoichiometry changes small. The default number of histories is set to 24. In case of a static simulation, the program choses typically 10 projectiles per processor core per history, so that with 24 cores we get 240 projectiles per history.
3. The number of processor cores. A modern workstation has typically 24 cores or more. A standard desktop PC has 4-8 cores available.

Typically the total number of projectiles is defined in the input script file. This ensures that the simulation of different computers reaches the same statistical level. The number of cores is fixed by the processor and the projectiles per history a selected automatically.

1.2 Parallel processing

Both programs utilize the message passing interface (MPI) routines [7], which allow parallel processing on computer clusters or computers with several processor cores. On a standard desktop PC (up to year 2022) one can use up to 8 processor cores in parallel, desktop workstations provide up to 64 processor cores. In contrast, other Monte Carlo BCA programs like SRIM or TRIDYN work “sequential”, i.e. only use a single processor core. In dynamic mode, parallel processing can be done only for the number of ions per history, because a target update is required after each completed history.

In IMINTDYN Version 8 we have implemented MPI file write routines to create efficiently data and trajectory output by writing collision data from each processor to a single output file. This is done using a nonblocking write command to a shared file pointer (in append mode pointing to the end of file). The routine MPI_WAIT ensures that the write process is thread safe and no data are lost during a write process. Computing time for data writing is now negligible. In previous IMINTDYN Versions and also in SDTrimSP, the conventional data writing procedures are a severe bottleneck an may be responsible for up to 50% of the required computing time.

MPI_Bcast routines were extended to allow comfortable broadcasting of character array, and logical vectors between processors.

Obsolete compilation and program modes:.

- The Sequential mode, which was optionally used by SDTrimSP and earlier versions of IMINTDYN, has been removed in Version 8.0
- The Compiler option for 32 bit operating systems has been removed in Version 8.0
- SDTrimSP had an option to restart a simulation if a previous simulations stopped due to a computer hardware failure (obvious a common issue when working with HPC computer clusters). This option requires to write all simulation parameters and variables to a file after every history, which is a severe bottleneck increasing the computation time. With new workstation processors such as intel i9 or AMD Ryzen Threadripper the typical computation times do not exceed several minutes or several ten minutes for complex simulations. A backup to restart a simulation is therefore obsolete an is no longer used in IMINTDYN.

1.3 Comparison of the parallel processing efficiency with SDTrimSP

In the manual for SDTrimSP Version 6.00 [4] we find some results for the parallel processing efficiency. Tables 2 and 3 in ref. [4] show the execution time as function of the number of processors used for a model case of 1 keV Fe into TaC impinging at normal incidence. The simulations with SDTrimSP use 20000 histories and 512 projectiles per history and were tested on an IBM Regatta 1.3GHz with up to 512 processors as well as a Linux/Intel cluster 2.8GHz with up to 64 processors. Our simulations were tested (i) on a Desktop PC with intel i7-2700 3.4GHz processor with a Linux virtual box using 4 processor cores, and (ii) on a DEL Precision workstation equipped with an AMC Ryzen Threadripper 7965WX 4.2 GHz processor with 24 cores and 48 threads [8].

Figure 1 shows the result of the comparison. The AMD Ryzen 7965WX using 24 cores finished the simulation after 3 minutes and even in 2.6 minutes is all 48 threads are used. This is the parallel efficiency of the IBM Regatta with 512 cores 1,3GHz and two cores per chip. The desktop PC with 4 cores needs about 102 minutes (marked with a cricle), comparable to the fastest time of the Linux cluster with 4 cores. The significant time differences between static and dynamic simulations are almost absent in the IMINTYN simulations. This comparison shows that (i) the benchmark tests described in the SDTrimSP Version6.00 document are based on rather old computers (around year 2001) and (ii) an up to date single workstation using an AMD Ryzen or Intel i9 processor can finish such a simulation in 3 minutes or even less. The todays high-end AMD processor has 64 cores and would need only about 1 min 10 sec to complete the simulation.

Most extensive simulations like ion scattering with millions of projectiles (RBS, C-ERDA, LEIS) or simulations of pattern formation requiring up to 18 angular steps for up to 9 different surface curvatures can be done in about 20-30 minutes on the workstation. This is typically much faster than the time to perform the real experiment.

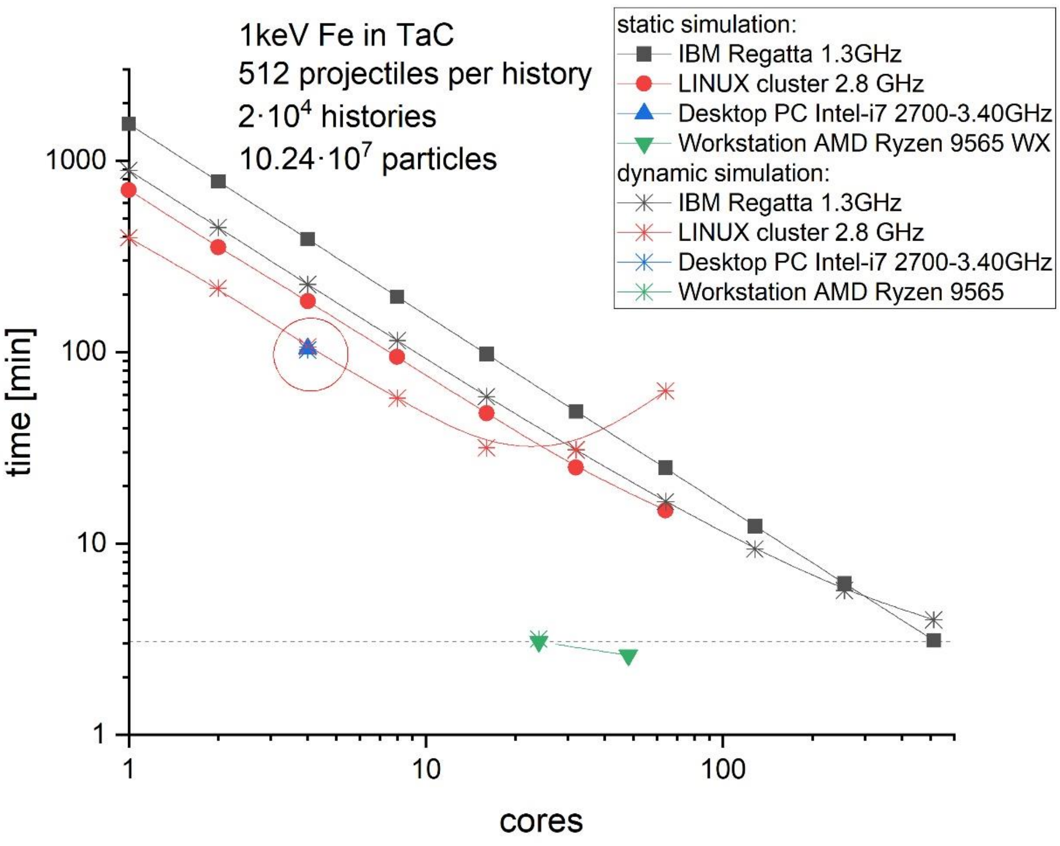

Fig.1 Performance and Parallel efficiency of IMINTDYN8

For an efficient use of the features of IMINTDYN we only need a state of the art desktop workstation with an Intel i9 or an AMD Threadripper processor and preferably the intel oneAPI Compiler. All together an invest of less than 10,000 € in 2023.

1.4 Simultaneous weak collisions:

Whereas the TRIM and SRIM codes use the pure binary collision approximation, TRIDYN, Trim.SP, SDTrimSP as well as IMINTDYN use the concept of simultaneous weak collisions. This concept was introduced by Biersack and Eckstein in 1984 [9] to improve the accuracy of sputter yield predictions. Instead of considering a single binary collision with maximum impact parameter determined by the local atomic density, additional weak binary collisions take collisions with next nearest neighbors into account. Typically, three simultaneous collisions are sufficient to obtain a complete collision cascade scenario. The concept is relevant for low energy collisions below about 10 keV. Neglecting simultaneous weak collisions results in wrong sputter yields, increased ion ranges and incomplete collision cascades.

1.5 Simulation limitations of SDTRIMSP

Due to the program code structure SDTRIMSP simulations can only handle up to 9 target and projectile species. SDTRIMSP was designed to simulate low energy collisions and only uses so-called monolayer collision steps (an option also available in SRIM) an only electronic stopping calculation for the low energy regime were implemented. Only the recent version 6 of SDTRIMSP provides electronic stopping calculation using the Ziegler-Biersack stopping formulas suitable for MeV projectile energies. SDTRIMSP and also TRIDYN are command line programs which require a script file to define a certain simulation. These scripts are often hard to read ASCII files. TRIDYN uses a script file containing only undocumented numeric data. SDTRIMSP uses more structured script files but with hard to understand abbreviations of parameters (for example: idrel= 0, +/-1 to define a dynamic simulation 0 or a static simulation +1 or a static simulation without calculating the recoil cascade -1).

1.6 New features of IMINTDYN

Compared to SDTRIMSP the new software IMINTDYN provides many additional features.

- **More than 10 target and projectile species** (technically up to 999) can be defined.

  Due to the program code structure of TRIDYN and SDTrimSP simulations can only handle up to 10, respectively 9 target+projectile species. TRIDYN before version 2017 is restricted to only 4 target+projectile species. IMINTDYN can now handle up to 999 target+projectile species. This allows the definition of quite complex target multilayer structures which can be simulated in dynamic mode. The same target element species can appear more than once, so that identical target elements belonging to different target layers can be followed individually. Concentration versus depth profiles as output of a simulation can easily be used as input for a new simulation, for example to simulated the SIMS sputter profile for a complex target structure as a result of previous ion irradiation.

  - Several isotopic target and projectiles species are defined, such as ^{6}Li, ^{7}Li, ^{10}B, ^{57}Fe
  - Tabulated data for gases include densities for solid, liquid and gas states as well as some isoptopes.

- As complementary models for low energy ion solid-interactions, in particular to simulate sputtering processes, one can chose between the common **surface binding energy (SBE) model** and the novel **bulk binding energy (BBE) model** [3].
  - The SBE model assumes a completely elastic collision cascade and a planar attractive surface potential which defines an energetic threshold for sputtering. This threshold is based on sublimation enthalpies of the target species. A constant energy loss for each created recoil can be defined as bulk binding energy, but this value is recommended to be zero. The cutoff energy to finish an event in a collision cascade is set to an arbitrary value between typically 0.1 eV up to displacement energies of more than 20 eV. The SBE model essentially decouples the processes in the collision cascade form the sputtering process.
  - The bulk binding energy uses binding energies based on sublimation enthalpies for each recoil in a collision cascade. The energy is consumed if a recoil is generated and released if a particle is stopped inside the target. A planar

surface potential no longer exists. The cutoff energy is derived from sublimation enthalpies and is taken as 33% of the sublimation energy of an atom. The BBE model directly couple the collision cascade processes to sputtering and the cutoff energy is no longer a free parameter. The BBE model gives reliable sputter data for many tested ion-target systems.

- The program calculates the **crater function moments** for erosion craters, implantation craters and recoil distribution craters up the 6th order. It is also possible to introduce a weak surface curvature to calculate the direct curvature dependent of the crater functions, which is needed for analytical theoretical model of ion induced pattern formation [10].
- Similar to molecular dynamics simulation programs, the local energy density, represented as **local temperature**, is dynamically calculated from the local deposited nuclear energy loss. In this way it is possible to follow up the initial sample conditions far from thermodynamic equilibrium. It is also possible to use modified sublimation enthalpy values in case of local high temperatures in the collision cascade.
- **The displacement energy** is usually set to zero and a non-zero value is only used to speed up simulations, if details of the recoil cascades can be neglected.
- **Flexible mean free path selection** from mono-layer collision steps to large mean free path based on a definable maximum angular and energy straggling in a collision. This enables fast simulation for incident MeV projectiles.

  SRIM [13] allows simulations for projectiles over a very broad range of energies. The fastest but less accurate modes are “Ion ranges and quick calculation of damage” or “ Detailed calculation with full damage cascades” and uses a large mean free path between subsequent collisions. The mean free path is chosen so that relative electronic energy loss and angular straggling between collisions remain small, typically less than 5% or 5°. The SRIM mode “Monolayer collision steps” is also used exclusively in TRIDYN and SDTRimSP and uses a fixed mean free path based on the average spacing of target atoms. The mode is most accurate but very time consuming at higher energies. IMINTDYN allows to chose the mean free path by selecting the desired accuracy regarding relative electronic energy loss and angular straggling between subsequent collisions. Whereas the application of TRIDYN and SDTrimSP is limited to low energy collisions, IMINTDYN allows fast and efficient simulations also for high projectile energies up to 2 GeV.

- Selection of **different electronic stopping models**.

  TRIDYN [11, 12, 14] and SDTrimSP up to version 5 only use theoretical models for electronic stopping based on theories by Lindhard and Scharff (non-local stopping theory) and Oen and Robinson (local stopping theory). Since Version 6 SDTrimSP optionally uses the Ziegler-Biersack stopping power formulas, which is based on a fit to experimental data for H and He stopping and applying scaling laws for heavier target elements and effective projectile charges. SRIM uses also stopping power formulas fitted to experimental data and can be seen as an upgrade to the Ziegler-Biersack stopping power formulas. IMINTDYN allows selection of all these stopping power models, including the SRIM-2013 stopping power data.

- Lindhard-Scharff (LS) and Oen-Robinson (OR) for the low energy regime.
- Ziegler-Biersack stopping formula for a broad energy regime up to 2 GeV
- SRIM-2013 electronic stopping data taken from SRIM's SR module and converted to a table of stropping data for all elements. The program offers a complete set of 92x92=8464 datasets for all projectile-target element combinations and an energy range up to 2 GeV.

- **The vacancy is introduced as a regular possible target species**, not only as a counter variable (as is the case in SRIM, SDTRIMSP and TRIDYN). Simple models were developed so that vacancies can be generated and also annihilate. With vacancies as a target species it is possible to simulate van-der-Waals separated atomic layers (such as graphene) or porous materials (like ion irradiated Ge). The behavior of a vacancy containing porous material with a given atomic density is different to one without vacancies but with a reduced atomic density.

  Previous BCA codes allow prediction of damage formation and vacancy production bases on the Kinchin-Pease model, the use of displacement energies and a book-keeping counter for potentially vacancy generating collisions. The vacancy, however, does not appear as additional target element. IMINTDYN now allows to define the vacancy as target element "Vac" equivalent to other target elements like "Si" or "Au". The vacancy as atomic weight zero and atomic number zero. It does not contribute to scattering or recoil but only increases the mean free path until the next collision with a massive target element. The vacancy occupies an atomic volume equivalent to the atomic volume of neighboring massive target elements. Vacancies can be either predefined or are dynamically generated in projectile target atom collisions. Vacancies can also annihilate if a massive projectile or recoil is stopped in the close vicinity. Generation and annihilation are described by various selectable models for the probability of such processes. In this way it is possible to simulate systems which dynamically develop a porous structure. It is important to notice that introducing vacancies is NOT equivalent to reduce the overall density of a target.

  The vacancy as a target atom is very valuable as spacer layer between individual graphene monolayers, and space-holder for the position in the center of a 6-ring atomic structure like graphene or other hexagonal structured materials. In this way, two-dimensional materials or polycrystalline hexagonal structures can be more accurately represented by an amorphous structure. For example, a graphene layer would consist of 33.33 % vacancies (free volume in the 6 ring structure) and 66.67 % carbon atoms with a corresponding factor 1.5 higher overall virtual atomic density (including vacancies).

  If the space layers between graphite sheets is also included as layers of vacancies, then we have 1 graphite sheet and 2 vacancy layers. The virtual atomic density then increase to 3 x 1.5 = 4.5 times the graphite density.

- Prediction of **Dimer and Trimer sputtering yield** based on a simple thermodynamic model which uses the formation enthalpies of dimers and trimers rather than the elemental formation enthalpies. The probabilities for sputtering of dimers and trimers depends on a Boltzmann distribution, normalized with a partition function and also the local elemental concentrations at the surface of a target. The option “dimer_sputtering” uses a table of compound formation enthalpies and explains the experimentally large sputter yields for e.g. erosion of carbon targets and also some oxide targets like SiO2, Ta2O5 and BaO. This option allows prediction of correct sputter yields without using the surface binding energy as a fit parameter (as it is often done in SDTrimSP).

- **Enforced scattering** is a new option to efficiently simulate ion scattering and elastic recoil collisions with high projectile energies. Enforced scattering dramatically reduces the impact parameter and selects only scattering events within a certain angular regime of scattering and recoil angles. In this case the actual scattering cross sections and scattering probabilities are followed up using a parameter linked to each scattered or recoiled particle.
- It is possible to use **non-Rutherford cross sections** taken from the IBANDL data base [16]. Now it is possible to simulate e.g. backscattering of MeV protons at light elements.
- The program provides **tables of the impact parameter versus scattering angle** for the interaction potentials used.
- The input script file now uses **easy to understand script language commands** an also comments to help setting up a simulation script.

SDTrimSP and also TRIDYN are command line programs which require a script file to define a certain simulation. These scripts are often hard to read ASCII files. TRIDYN uses a script file containing only uncommented numeric data. SDTrimSP uses more structured script files but with often hard to understand abbreviations of parameters (for example: idrel= 0, +/-1 to define a dynamic simulation 0 or a static simulation +1 or a static simulation without calculating the recoil cascade -1). IMINDYN uses script commands which are much more intuitive and are explained by comment lines in the script files.

IMINTDYN scripts allow the definition of series simulations for series of ion energies, incidence angles or fluence. The resulting output data are labelled with a series index and post processing programs automatically extract detailed distributions of scattered, stopped and recoiled atoms for each set of the series simulation.

- **A comfortable batch file is used to run simulations** with a sequence of script files and also automatically start post processing programs to create all kind of possible output data. Several post processing programs were written to create output for various ion beam analysis simulations such as elastic (non-Rutherford) backscattering (EBS), Elastic recoil detection analysis (ERDA), Coincidence-ERDA and elastic recoil coincidence spectrometry (ERCS), the latter technique analyses the energies of coincident scattering and recoil events.

- **The input files defining layered target structures, energy and angular profiles** are made more comfortable. A final target structure is taken to a new input file, which can directly be used to simulate SIMS sputtering profiles.

- **Book-keeping** of all involved particles is improved and allows easy tracking of coincident events and to follow up dual triple and multiple collision events.

- **Post processing programs** generate a variety of output data files on composition profiles sputter yield distributions, backscattering distributions etc. Post processing programs also generate simulated spectra for backscattering elastic recoil detection or coincidence scattering techniques.

- **The ASCI output files** are formatted in a way that an easy import into ORIGIN® software is possible

3. Some words on random number generators

IMINTDYN now offers three different options of random number generators. These options can be selected when compiling the program using the compiler options

1. RAND = RAND_INTEL
2. RAND=RAND_CRAY
3. RAND=NR_RANDQ1

Previous random number generators ART_RAND, RAND1 And RAND2 used in SDTrimSP are lo longer used. The description in the SDTrimSP subroutine random.f90 says:

“ART_RAND” is only mean for comparison purpose. The calculation only depends on number of histories and number of projectiles per history and results are not really random”

„RAND2 is only recommended if the compiler does not allow 4 byte integer. This random generator i not fully random distributed”. Today 4 bytes is the minimum standard and RAND2 is obsolete

NR_RANDQ1 is a random number generator describe in the book Numerical Recipies.. Programmed after ranq1-C++-code from Numerical Recipes, 3rd edition, chapter 7.3.1.

The period is : 1.8*10**19. Fortran-Code by Silvio Gori, Max-Planck-Institut für Plasmaphysik, Garching, Germany, adapted to SDTrimSP by U. von Toussaint

NR_RANDQ1 routine is started with a random seed _number and then creates a random number 0< r < 1 as well as a new seed number. Therefore, an initial seed must be specified only at the beginning for each processor. Specifying additional seeds in between a simulation can be done but is not necessary. The initial seed number is an integer(8) variable

RAND_CRAY can be used as reliable alternative. There are comments that the generator produced the value zero exactly.

RAND_INTEL is the built in random number generator as portablity function of the Intel Fortran Compiler. The algorithm used is a "Prime Modulus M Multiplicative Linear Congruential Generator," a modified version of the random number generator by Park and Miller in "Random Number Generators: Good Ones Are Hard to Find," CACM, October 1988, Vol. 31, No. 10. The seed number is interg(4) !!

The generators need to be initialized and then random numbers 0 < r<1 can be obtained

Use of RAND_INTEL:

Initialize and reseeding by r=random(iseed) with iseed > 2 and integer(4)

Restart and select 1st random number: r=random(1) . This always selects the same number !

Select next random number: r=random(0) in a series, which is not too long. The numbers repeat.

To initialize we calculate inter(4) seedclock using the system_clock subroutine and then set a seed number of each processor with pid identifier using integer(4) seedclock = seedclock+1737*pid. 1737 is the year where the University Göttingen was founded.

The series of random numbers is not too long and without re-seeding the same random number sequences may occur. Therefore, the initialization is repeated for every history.

4. Description of program features

4.1 Bulk binding energy model

As complementary models for low energy ion solid-interactions, in particular to simulate sputtering processes, one can chose between the common surface binding energy (SBE) model and the novel bulk binding energy (BBE) model.

The SBE model assumes a completely elastic collision cascade and a planar attractive surface potential which defines an energetic threshold for sputtering. This threshold is based on sublimation enthalpies of the target species. A constant energy loss for each created recoil can be defined as bulk binding energy, but this value is recommended to be zero. The cutoff energy to finish an event in a collision cascade is set to an arbitrary value between typically 0.1 eV up to displacement energies of more than 20 eV. The SBE model essentially decouples the processes in the collision cascade form the sputtering process.

The bulk binding energy uses binding energies based on sublimation enthalpies for each recoil in a collision cascade. The energy is consumed if a recoil is generated and released if a particle is stopped inside the target. A planar surface potential no longer exists. The cutoff energy is derived from sublimation enthalpies and is taken as 33% of the sublimation energy of an atom. The BBE model directly couples the collision cascade processes to sputtering and the cutoff energy is no longer a free parameter. The BBE model gives reliable sputter data for many tested ion-target systems.

4.2 Dynamic collision cascade temperature

The simulation is usually done for a fixed target temperature, which is by default room temperature. However, the huge nuclear energy loss as well as sublimation energies released when an atom is stopped inside a target may lead to rather high local energy densities in the collision cascade volume. These temperatures can reach several 10,000 K in certain cases and represents the initial temperature of a thermal spike.

IMINTDYN can follow up the local energy deposition and represent it as local temperature as well as Gaussian temperature distribution. In this way it is possible to evaluate a collision cascade, whether a thermal spike may be initiated or not. As a consequence of such a thermal spike we may consider e.g. surface diffusion. In addition, a locally high temperature may lead to a reduction of the sublimation enthalpy and thus increase the sputtering yield. This effect is discussed in the NIMB 2022 publication by Hofsäss and Stegmaier.

In the IMINTDYN code, we have implemented an option to calculate a local temperature dependent sublimation enthalpy based on the tabulated gas phase chemistry enthalpies. We propose that the high local temperature at positions where recoils are generated will leads to a reduced sublimation enthalpy and thus a higher sputter yield. However, comparison to experimental data shows that the standard sublimation enthalpies reproduce experimental sputter yields in many cases. One reason is that sputtered recoils are generated close to the surface, sufficiently far away from the center of the collision cascade. The average local temperature T_{AV} at these recoil positions is usually below 3000 K and has therefore little influence on the sublimation enthalpy. A temperature effect should be most pronounced in cases where T_{av} becomes high, for example for irradiation of Fe, Co or Ni with Xe ions at grazing incidence of about 70° and with energies between 3-5 keV. Here we expect values for T_{av} of more than 4000 K.

Instead of using the formation enthalpies at standard conditions, we can optionally use the lower enthalpies at the corresponding temperatures T_{av}. The enthalpy difference ΔH as function of temperature is in general obtained by integration over the heat capacity $C_P(T)$

$$\Delta H = \int_{T1}^{T2} C_P(T)dT \qquad (1)$$

In the simplest case we can approximate $C_P(T) = a + bT + c/T^2$ or assume $C_P = const$ applying the Dulong-Petit law. If no tabulated ΔH values exist, we may use eq. (1) to calculate it. A more detailed representation is given by the Shomate equation [15] (an established method to calculate thermo-chemistry data using polynomial equations), taking the condensed phase (solid and liquid) and the gas phase behavior into account. The NIST Chemistry WebBook SRD 69 provides data for Cp(T) and ΔH(T) up to T_2 = 6000 K for many elements and compounds.

Although the polynomial Shomate equations seems to be quite inconvenient, we use an approximate linear relationship

$$\Delta H(T) = \Delta H(RT) - \left(\Delta H(RT) - \Delta H_{6000K}\right)\frac{T-298K}{6000K-298K}, \tag{2}$$

for most elements and compounds and the tabulated gas phase sublimation enthalpy values H_0 for the elemental species to correct the standard state sublimation enthalpies in IMINTDYN.

The table-elements.txt contains data for the sublimation enthalpy ΔH_{6000K} at 6000 K, derived from the NIST database for gas phase chemistry. IMINTDYN uses the interpolated enthalpy according to eq.(2).

The table table-compounds.txt contains data for the sublimation enthalpy **change** of a binary compound from RT to 6000 K, derived from the NIST database for gas phase chemistry. IMINTDYN then calculates the formation enthalpy of a compound at a temperature 6000 K using

$$\Delta H_{6000K,compound}(T) = \max(0., \Delta H_{compound}(RT) - \delta(\Delta H_{6000K.compound})) \tag{1.3}$$

The temperature dependent compound formation enthalphy is then

$$\Delta H_{compound}(T) = \Delta H_{compound}(RT) - \left(\Delta H_{compound}(RT) - \Delta H_{6000K,compound}\right)\frac{T-298K}{6000K-298K}, \tag{4}$$

4.3 Electronic energy loss models

IMINTYN provides several electronic energy loss options

a) Oen-Robinson

This model referred to as local energy loss is a modification of the model by Firsov, who proposed a model where the electron clouds of the colliding atoms overlap, assuming a collision time long enough for complete mixing of the electrons in the overlap region.

OR stopping is based on formula 5.1.6 from the book of Eckstein. The overall energy dependence is proportional to $\sqrt{E}$ and can therefore only be used to sufficiently low energies.

b) Lindhard-Scharff

The electron gas in a solid behaves like a viscous medium for the moving ion. This was already assumed by Fermi and Teller when they derived a velocity-proportional stopping. Later, Lindhard and Scharff gave a formula for the inelastic stopping cross-section, referred to as continuous stopping or non-local stopping (see Eckstein eq.

5.2.1). The overall energy dependence of LS-stopping is proportional to $\sqrt{E}$ and can therefore only be used to sufficiently low energies.

c) Average of Oen-Robinson and Lindhard-Scharff

Since LS and OR stopping deviate at lower energies below about 10 keV, one often uses an average stopping cross section in SDTrimSP and TRIDYN. This option can also be used in IMINTDYN

$$S = \frac{1}{2}\left(S_{OR} + S_{LS}\right) \tag{1.5}$$

d) Ziegler-Biersack stopping data

Ziegler-Biersack introduced stopping data for all elements using scaling laws, effective projectile charegs and fit parameters for He and H experimental stopping data. See Eckstein page 70-72. Stopping is calculated using fit formulas with fit parameters are tabulated in files SCOEF95A and SCOEF95B. The advantage of the ZB Stopping data is the large accessible ion energy range for all projectile target combinations.

e) SRIM-2013 Stopping data

SRIM stopping data are similar to ZB stopping data but are based on a further developed data base of experimental stopping data. The fit parameter tables are significant larger compared to ZB tables and not really documented. Therefore, the use of SRIM-2013 stopping data is restricted to the SRIM program, or one can use the program SRMODULE to extract a table of stopping data for selected projectiles and targets as well as an ion energy regime as well as different stopping units.

We used the SRMODULE program and the Stopping and Range feature of the SRIM package to calculate stopping data in the energy regime 10 eV up to 2 GeV in stopping units eV/(10^{15} at/cm^2). These units are independent of the atomic density. The ASCII output of SRMODULE consists of 218 values in logarithmic energy steps. Since the ASCI files are quite uncomfortable to use, we have calculated ASCII files for all projectile target combinations using a Python script and then converted the stopping data into 92 tables for each projectile element. Therefore, IMINTYN has a complete set of SRIM-2013 stopping data available. Each table consists of 219 rows of energies and 92 columns of stopping data for each target element. Stopping data for intermediate energies between tabulated values are determined by interpolation.

IMINTDYN reads the necessary stopping data files based on the atomic species defined in the input script file.

f) H stopping cross section

Identical to SDTrimSP and TRIDYN one can chose a stopping model for Hydrogen as projectile for energies between 1 keV and 100 MeV (see Eckstein, pages 69-70). The stopping model used fit functions for three different energy regimes.

Whereas SDTrimSP only uses a non-relativistic treatment, IMINTDYN also considers a relativistic correction, so that a divergence of the first log term in eq. 5.2.7 of Eckstein page 50 (which is the correct relativistic formula) does not occur. Now the stopping cross section works up to 2 GeV.

g) He stopping cross section

Identical to SDTrimSP and TRIDYN one can chose a stopping model for Helium as projectile for energies between 1 keV and > 10 MeV (see Eckstein, pages 69-70). The stopping model used fit functions for two different energy regimes.

4.4 Vacancy generation and annihilation

The vacancy was added as a new target element with atomic number Z = 0, Mass A = 0 and a selectable atomic volume. Of course, a vacancy cannot act as a projectile. A vacancy can be applied as an additional target component in static simulations. If a certain vacancy concentration is assumed, the overall mass density of the target is reduced but the maximum impact parameter p_{max}, derived from the constant) atomic density by $p_{\max} = 1/\sqrt[3]{n}$, remains unchanged. In contrast, reducing the mass density be reducing the atomic density n leads to a slightly larger value of p_{max}. As a consequence, scattering with larger impact parameter or small scattering angle (forward scattering) is enhanced. This indicates that inserting vacancies has a different effect than reducing the density. As a result, depth distributions and sputter yields will differ in both cases.

A collision with a vacancy has no effect at all, which means the path of a projectile continues with deflection angle 0° and no further energy loss.

If a collision with a target atom takes place and a recoil is generated with recoil energy $E_{recoil} > E_{displ}$, then the program has to decide if a vacancy is generated. In the simple Kinchin-Pease model implemented in SRIM, TRIDYN and SDTrimSP such a collision simply increases a vacancy counter. This leads to output values such as vacancies per ion. Typically, in SDTrimSP a recoil is generated without creating a vacancy. This leads to an undercoordinated local structure with a reduced atomic density. In case of a dynamic relaxation, this density relaxes towards the initial density and leads to a relaxation of the layer thickness and a modification or adjustment of the layer composition.

The formation of a vacancy usually requires an activation energy for vacancy formation. This energy may be compared with thermal energies and an equilibrium vacancy concentration n_{vac} can be determined from

$$n_{Vac} = n \cdot exp(-\frac{E_A}{kT}) \tag{1.6}$$

For an activation energy of E_A = 1 eV and a typical atomic density of n = $5 \cdot 10^{22}$ at/cm^3, we find a thermodynamic equilibrium vacancy concentration at room temperature of only n_V = $8 \cdot 10^5$ vac/cm^3. In an amorphous material a vacancy is not defined. Instead one uses the concept of undercoordinated bond structures. In certain amorphous solids one may also consider vacancies as empty sites nearby dangling bonds. Often the amorphous structure is represented by ring structures of bonded atoms, with a variation of atoms forming a ring. A typical example is amorphous carbon with e.g. 5, 6 and 7 fold rings. The 6 fold ring with 6 carbon atoms would represent the basic structure in graphite.

In the IMINTDYN software, a vacancy can be either defined as a static component of a target or vacancies can be created or annihilated dynamically. In a dynamic simulation, we tried to implement the dynamic formation and annihilation of vacancies within the process of a collision cascade. This is different to usual models and theories of vacancy formation and annihilation, because we are considering here a system extremely far from equilibrium. The energy needed to create a vacancy by recoiling a target atom is supplied by the projectile kinetic energy.

In our approach, a vacancy is generated with a certain probability. The formation energy is provided in the collision event by the projectile kinetic energy. The probability p_{vac} is calculated based on a general formation probability q_{vac} with $0 \leq q_{Vac} \leq 1$ and the local vacancy concentration c_{vac} = qux (mm,ncp_vacancy) in layer mm for the target element with index ncp_vacancy. c_{vac} is a fraction between 0 and 1. The probability for vacancy formation is given by selectable functions for each massive target species with index n $\in\{1,\ldots,ncp\}$. Some functions reduce the probability p_{vac} with increasing c_{vac}, others let p_{vac} increase towards a value 1 with increasing c_{vac}.

mode 0:	$p_{vac} = q_{vac}$	constant
mode 1: (default)	$p_{vac} = q_{vac} \cdot (1 - c_{vac})^2$	quadratic –
mode 2:	$p_{vac} = q_{vac} + (1 - q_{vac}) \cdot \left(1 - (1 - c_{vac})^2\right)$	quadratic +
mode 3:	$p_{vac} = q_{vac} \cdot (1 - c_{vac})$	linear -
mode 4:	$p_{vac} = q_{vac} + c_{vac}(1 - q_{vac})$	linear +
mode 5:	$p_{vac} = q_{vac} \cdot (1 - c_{vac})^3$	cubic –
mode 6:	$p_{vac} = q_{vac} + (1 - q_{vac}) \cdot \left(1 - (1 - c_{vac})^3\right)$	cubic +
mode 7:	$p_{vac} = q_{vac} \cdot \frac{1}{2}\left(1 + \sin(\pi(c_{vac} + 0.5))\right)$	sine -
mode 8:	$p_{vac} = q_{vac} + (1 - q_{vac}) \cdot \frac{1}{2}\left(1 - \sin(\pi(c_{vac} + 0.5))\right)$	sine +

Other mode values are identical to mode 0.

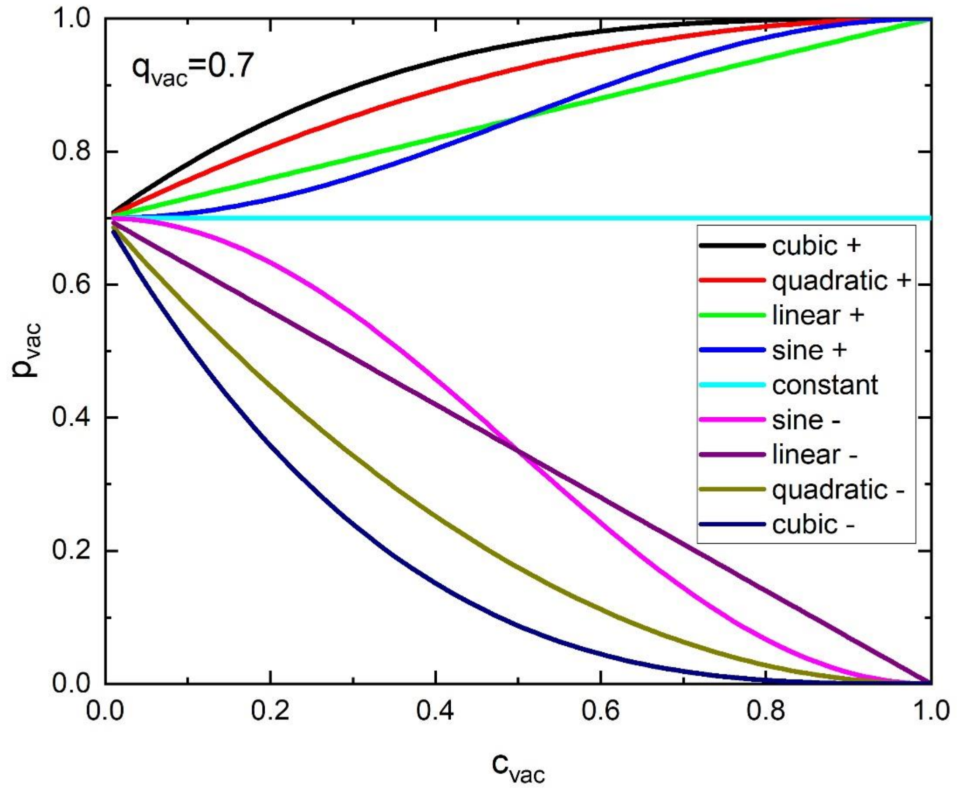

Figure 2: probability p_{vac} of vacancy formation as function of the local vacancy concentration c_{vac} for different model assumptions and initial probability q_{vac} = 0.7.

It is assumed in some modes that the vacancy formation depends on the local vacancy concentration. For a very low vacancy concentration, it is likely that the local structure does not relax and a vacancy is created. Therefore, in the extreme case of q_{Vac} = 0, we get p_{vac} = 1 or $p_{vac} = q_{vac}$. For higher local vacancy concentrations, a relaxation without vacancy formation may be more likely, because of the larger open volume available for relaxation and an increases inner surface tension. It may also be possible, that void formation is favorable and p_{vac} approaches 1 as c_{vac} increases. q_{vac} describes the initial probability for c_{vac} = 0.

If a projectile or recoil is stopped, then the stopped particle may annihilate with a vacancy. This annihilation of course must depend on the local vacancy concentrations. We introduce a bond coordination number n_{coord} to count the nearest neighbors of an arbitrary atom site. If one of these neighbor atoms of a particle stop position is a vacancy, then the vacancy is annihilated. Therefore, the probability of vacancy annihilation is given as

$$p_{annihilation} = \min\left[1, n_{coord} \cdot c_{vac}\right] \tag{1.7}$$

As an example, if the local vacancy concentration is q_{Vac} = 0.1 and the coordination number is `(i_vac_coord)` n_{coord} =4, we find an annihilation probability of $p_{annihilation}$ = 0.4 or 40%.

Typical settings are:

```
vacancy_mode =1
q_vacancy =0.7,0.1,0.0,0.7
i_vac_coord = 4  !( 3-fold coordination: 3)
```

For surface near regions, the vacancy concentration may drop due to out-diffusion of vacancies. This can be modelled by using the logical command (logical vacancy error function) lvacerf =.true. . In this case, an error function centered at a given target slice from the surface and a with given in units of target slices can be defined. The following commands

lvacerf=.true. default: lvacerf=.false.
vacerf_sigma = 3
vacerf_pos = 6
qumax = 0.3 (for vacancy target species)

limit the maximum bulk vacancy concentration of 0.3 near the surface using an error function centered at slice 6 and width 3 slices. An example of the error function profile is given in the following figure.

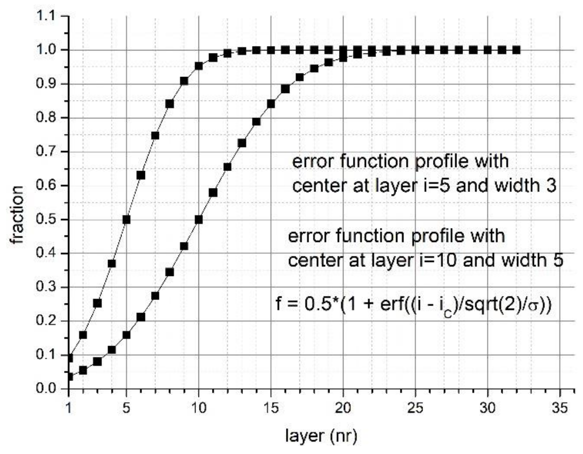

Figure 3: error function profiles at the surface for vacancy annihilation.

Note: Vacancies cannot be sputtered off. Therefore, if vacancies are used to fill up empty sites in a structure like graphene or metal-dichalcogenides, one must set lvacerf=.true. to allow vacancies to be annihilated (emitted into vacuum) near the surface. Otherwise the surface recession may get wrong and one observes an artificial growth of the target.

Vacancies will increase the mean free path of a projectile and thus lead to an increased ion range. This is in particular important for a heavy projectile and a light target matrix.

Vacancies, however, are not free of electrons and a projectile passing through the volume of a vacancy will experience electronic stopping. The program can include elektronic stopping in two ways.

a) One can specify certain target elements which may be replaced by vacancies using the command

Vacancy_species =”Si”, “C”, etc.

The program the uses the average electronic stopping for these elements based on the global atomic concentrations. In addition a scaling factor can be defined, which is by default 1.0

vacancy_stopping_fraction = 1.0

b) If no elements are specified as shown in a), the program uses the element with highest global atomic concentration to set the vacancy electronic stopping.

Experiments with HR-RBS after implantation of W into a-C have shown that a value of vacancy_stopping_fraction = 1.0 is a good choice. Furthermore, the atomic density of a vacancy has little influence. A low atomic density leads to larger ion ranges, however, a low atomic density leads to a low electron density, so there is no visible effect when analyzing an implantation profile with ion beam analysis.

Vacancies have a much stronger effect on ion ranges than a scaled redcution of the overall atomic density n. The latter increases the mean free path of a projectile by $n^{1/3}$, the vacancy increases the mean free path between atomic collisions by factor of 2.

4.5 Use of tabulated data

IMINTDYN provides a number of tabulated atomic data. These can be used in most cases, but care must be taken regarding atomic densities and formation enthalpies.

The **table-elements.dat** contains atomic data of the elements, including sublimation enthalpies, masses, mass densities and atomic densities.

Note: For gases the densities are given for the gas, liquid and solid phase at normal pressure. However, gases and some other elements like alkali atoms (Li,Na,Rb,Cs) or halogen atoms (F, Cl, I) incorporated into a solid will have much higher atomic densities, in some cases up to 5-10 times the atomic densities of the elemental materials. In this case it is recommended to use the atomic density of the host solid or at least a typical atomic density of a solid like Si (0.05 at/$Å^3$). Otherwise implantation of these elements will lead to an artificial swelling of the target, which is reflected in a negative surface recession (growth of the target thickness), In particular noble gases may even form bubbles is solids with a density corresponding to the liquid or solid density at room temperature and thus extremely high pressure. The atomic density is then comparable to typical solids.

As a common 2D target element we have included the elements a-C1D and Vac1D use to model graphene as 3-layer system with 2 vacancy layers and 1 carbon+33% vacancy layer with an atomic + vacancy density 4.5 higher than the graphite density.

The **table-compounds.txt** contains data for binary compounds and can only be used if the target is binary compound irradiated by a different ion species.

The **table-dimers.txt** contains formation enthalpy data for dimer and can is used if dimer_sputtering=.true. is set. In addition a Dimer listed in the table must be specified and the individual elements should be present in the target, Example: dimer_compound="SiO" if the target is "SiO2".

The **table-isotopes.dat** contains data of isotopes of different elements. If the command

```
use_isotopes =
```

is used, then the list of target elements is expanded for each isotope specified (up to 8 for the case of Sn).

Other tables used for H and He stopping power calculation:

Table-HeStopping1.txt, table-heStopping2.txt, table- Hstopping.txt

4.6 Cutoff energies

The cutoff energy determines the lowest energy a moving particle can have before it is stopped. SRIM say nothing about the cutoff energy, but it is typically around 1-3 eV. TRIDYN and SDTrimSP allow a free choice of the cutoff energy. TRIDYN in particular recommends a value of 0.1 eV. However, such a low values leads to artificially large ion ranges in many cases. For example, the implantation of a heavy ion into a light matrix like P,Se,W into a-C leads to collisions with very low scattering angles mainly in forward direction. Reduction of the cutoff from 1 eV down to 0.1 eV implies about 10 more collisions with a mean free path of the atomic spacing (about 2 Å). This simple choice of the cutoff energy increases the ion range by about 2 nm. For 10 keV P,Se,W ion implantation into a-C, the mean ion range is around 8-10 nm and increases to 10-12 nm, i.e. by 25%, just due to the choice of a very low cutoff energy.

Therefore, the cutoff energy should be chosen carefully and it is usually not a free parameter. INIMTDYN uses settings for the cutoff energy, which differ from TRIDYN and SDTrimSP and also SRIM.

- In bulk binding energy mode, a cutoff-fraction of f_{cutoff} =0.333 as default is defined. The cutoff energy is then calculated from the elemental sublimation enthalpy E_{subl} as

$$E_{cutoff} = f_{cutoff} \cdot E_{subl} \tag{1.8}$$

The value f_{cutoff} is a default value and the cutoff energy is not a free parameter.

- In surface binding energy mode, the program also use a cutoff energy as a fraction of the surface binding energy similar to the BBE model. This option is set as default or is set using the command

```
cutoff_from_sbe = .true.
```

- For historical reasons, the file "table-elements.dat" has a column with cutoff energies. As default value IMI)NTDYN uses a tabulated cutoff energy of 1 eV in "table-elements.dat". This also includes gases and noble gases. However, this column is now obsolete.

- If individual cutoff energies for each element can still be specified. Then one sets

```
cutoff_from_sbe = .false.
```

  and specifies the cutoff energies

```
e_cutoff =  0.1, 0.2 ,0.3, 0.1, .....
```

  A zero value must not occur. If values of zero are specified, then a value of 1.0 eV is chosen as default. In case of a very low cutoff energy, e.g. 0.1 eV the computation time for a simulation can increase significantly.

4.7 Relativistic corrections

Relativistic corrections become important for very high projectile and recoil energies. The program takes relativistic corrections into account, if the Lorentz-factor γ of the moving particle is larger than γ = 1.01. This is checked at the beginning of every projectile and recoil loop. In this case the mass m of the moving particle is described as $\gamma \cdot m_0$ and the center of mass energy, the mass transfer and the mass ratio as well as a correction factor for electronic energy loss calculations is re-calculated accordingly. The energy straggling due to electronic energy loss uses the relativistic formula according to the SRIM book in chapter 6 eq. 6-18. If within a loop the relativistic correction does not become necessary because the moving atom energy is too small, then the correction is switched of for the rest of the loop.

4.8 Mean free path calculation

The program provides three options to determine the free path length between two subsequent collisions.

a) Monolayer collisions steps. This is identical to the SRIM option and is the only option available in SDTrimSP and TRIDYN. Here, the mean free path length is calculated based a cylindrical volume containing one target atom with cylinder radius p_{max} and

$$V = \frac{1}{n} = \pi p_{\max}^2 \lambda_0 \tag{1.9}$$

and local atomic density n.

$$\lambda_0 = n^{-1/3} \tag{1.10}$$

b) The so-called gaseous model, where the free flight path may vary slightly. This is explained in the Eckstein book on page 84-85. The free flight path is determined by λ_0 and a random number 0<R<1 defined by

$$1 - R = 1 - \exp(\lambda_{gaseous} / \lambda_0)\,. \tag{1.11}$$

$$\lambda_{gaseous} = -\lambda_0 \ln(R) \tag{1.12}$$

The free flight path varies between a value zero and infinity.

c) Variable free flight path. For higher projectile energies in the MeV energy range, the free flight path can be defined based on a minimum acceptable angular straggling end energy straggling. This option is also available in SRIM as default quick calculation mode. In SRIM the minimum acceptable angular deflection between two collisions is assumed to be less than 5° and the minimum relative electronic energy loss is assumed to be less than 5%. SDTrimSP and TRIDYN only have the monolayer collision step option and are therefore not suited to simulate efficiently high energy projectile cascades.
   In IMINTDYN has a variable free flight path option, where the accuracy can be selected. The script parameters available are:

```
free_flight_path =.true.
ffp_accuracy=0.01 (default)
```

   The free flight path accuracy define the maximum acceptable angular deflection between two collisions, where 0.01 corresponds to 1°, as well as the maximum acceptable relative electronic energy loss ΔE/E.

   The free flight path based on angular deflections is discussed in the SRIM handbook page 7-9. It assumes that the mean angular deflection per path remains constant. The typical mean deflection angle < 5° due to nuclear collisions with stopping cross section $S_{nucl}(E)$ is

$$\frac{\Delta E_{nucl}}{E} \le \sin^2(\vartheta_C) \tag{1.13}$$

   And the corresponding free path λ_{SN} is

$$\lambda_{SN} \le \frac{E}{S_{nucl}(E)\cdot n} \sin^2(\vartheta_C) \tag{1.14}$$

   Expressing everything in reduced values for energy and cross section

$$E = \frac{Z_1 Z_2 e^2 (M_1 + M_2)}{4\pi\varepsilon_0 a M_2} \varepsilon\,. \tag{1.15}$$

$$S_{nucl}(E) = \frac{\pi a^2 4 M_1 M_2 E}{\varepsilon (M_1 + M_2)^2} S_{nucl}(\varepsilon)\,. \tag{1.16}$$

$$\lambda \le \frac{E}{S_{nucl}(E)\cdot n} \sin^2(\vartheta_C) = \frac{1}{4\pi a^2 n} \frac{(M_1 + M_2)^2}{M_1 M_2} \frac{\varepsilon}{S_{nucl}(\varepsilon)}\,, \tag{1.17}$$

   with screening length a. We now use the empirical expression from Universal reduced nuclear stopping (see eg. 2.18 from SRIM Handbook)

$$\lambda_{SN} \le \frac{E}{S_{nucl}(E)\cdot n}\sin^2(\vartheta_C) \approx \sin^2(\vartheta_C)\frac{2}{\pi a^2 n}\frac{(\varepsilon^2+0.2\varepsilon^{1.5})}{\ln(1+\varepsilon)} \quad ;\varepsilon < 30 \tag{1.18}$$

$$\lambda_{SN} \le \frac{E}{S_{nucl}(E)\cdot n}\sin^2(\vartheta_C) \approx \sin^2(\vartheta_C)\frac{2}{\pi a^2 n}\frac{\varepsilon}{\ln(\varepsilon)} \quad ;\varepsilon > 30 \tag{1.19}$$

Example: 2 MeV H in TiH_2. The monolayer free path is 1.94 Å, the free path length of ffp_accuracy=0.01 (1°) is 100 Å and for ffp_accuracy=0.05 (5°) is 2000 Å. This will speed up a simulation such as forward or backscattering dramatically.

The free flight path λ_{SE} based on electronic energy loss along the path assumes that the energy loss shall not exceed a certain fraction of the kinetic energy E at the beginning of this path. The typical approach in SRIM is

$$\frac{S_{el}(E)\cdot n\cdot\lambda_{SE}}{E} \le ffp\_accuracy \tag{1.20}$$

$$\lambda_{SE} \le f\frac{E}{S_{el}(E)\cdot n} \tag{1.21}$$

λ_{SE} and λ_{SN} are calculated prior to each collision and the minimum value is chosen for λ .

$$\lambda = \min(\lambda_{SN},\lambda_{SE}) \tag{1.22}$$

4.9 Enforced scattering

Enforced scattering was introduced to efficiently simulated light ion scattering with MeV ion energies. The program provides different options for the most common scattering experiments using the command

```
Scattering_mode = "ebs" , or: "erd","cerda","ercs","nra"
```

- Elastic Backscattering (EBS,RBS)
- Elastic Recoil Detection (ERD)
- Coincidence Elastic Recoil Detection (C-ERDA)
- Elastic Recoil Coincidence Spectrometry (ERCS)
- Nuclear Reaction analysis (NRA)

Large angle scattering is often very unlikely for high energy light ions because the cross section proportional to $1/E^2$ and Z_1^2 , with projectile energy E and atomic number Z_1. The list of collision events like backscattered ions, forward scattered ions forward sputtered recoils consists mainly of very small angle collision events.

To circumvent this problem, IMINTDYN provides the enforce-scattering option. First, the program determines the average number of collisions navg of a projectile passing through the target or being backscattered. Then one selects a number num_scattering , how many large angle collisions per projectile shall occur. Calculate a reduction factor pmax_coeff for the maximum impact parameter pmax. Use a random number to find a probability p = num_scattering / navg. In p cases, the maximum impact parameter is reduced by factor pmax_coeff for a selected target atom. To find a proper value pmax_coeff, one can first calculate tables of scattering angle vs pmax. The program calculates pmax_coeff using a range of scattering and recoil angles for various scattering geometries.

```
enforce_scattering = .true.
single_scattering = .true. (recommended default)
enforce_rec_scatt = .true.
num_scattering = 1.0      (recommended default)
scattering_mode = 'none', 'cerda', 'ercs, 'ebs' ,'nra'
                   or 'CERDA','ERCS,'EBS','NRA'
scatt_species(1:num_species) = ...
cerda_angle = 0.    default
cerda_angle2 =90.   Default ! only forward swcattering possible
ercs_angle =0.      default
ercs_angle2 = 90.   Default  ! only forward scattering possible
ebs_angle =0.       default
ebs_angle2 = 180.   default
erda_angle= 90.     Default  ! onlöy backscattering/recoil possible
eerda_angle2 = 180. Default
```

The scattering mode angles define the angular regime for a large angle collision at a given random depth with a specific target atom. In addition we need to define the emission angle either in forward direction (transmission) or backward direction (backscattering, backward erosion) and for projectiles p and recoils r.

```
out_angle_pt =
out_angle_pb =
out_angle_rt =
out_angle_rb =
```

If enforce_rec_scatt =.true., then enforced scattering also applies to recoils. This option is only used in C-ERDA simulations like H-H- scattering or Li-Li scattering and only when single_scattering = .false.

If single scattering =.true., then a random number r is chosen to find a target position x_s = r·t_target/ffp where enforced scattering takes place. Only one scattering event is considered. In case of .false. plural scattering mode is active. The probability for enforced scattering p = num_scattering · ffp/t_target. A random number r is chosen and if r < p , then enforced scattering is activated. There may be more than one scattering event, so that plural scattering takes place. With proceeding path step n, the probability for a projectile to survive is $(1-p)^n$. This option is for test purposes only and may lead to modified scattering probabilities.

[redacted] Cross section amplification for enforced scattering

In general, the impact parameter is chosen from a random number

$$p_{impact} = p_{max} \cdot \sqrt{r} \quad ; \quad 0 \le r \le 1 \tag{1.23}$$

The average of $(p_{impact})^2$ is:

$$\left\langle p^2{}_{impact} \right\rangle = \left(\frac{\int_0^1 p_{max}\sqrt{r}\,dr}{\int_0^1 dr} \right)^2 = p^2{}_{max} \cdot \left(\frac{\int_0^1 \sqrt{r}\,dr}{\int_0^1 dr} \right)^2 \quad ; \quad 0 \le r \le 1$$

$$\left\langle p^2{}_{impact} \right\rangle = p^2{}_{max} \cdot \left(\frac{\frac{2}{3}\left(1^{3/2} - 0^{3/2}\right)}{1} \right)^2 = \frac{4}{9} p^2{}_{max} \,. \tag{1.24}$$

If p_{impact} is chosen between two limits $p_1 = f_1 \cdot p_{max}$ and $p_2 = f_2 \cdot p_{max}$, and $f_1 < f_2$ with p_1 is the larger impact parameter and p_2 the smaller one, then

$$\left\langle p^2{}_{impact} \right\rangle = p^2{}_{max} \cdot \left(\frac{\int_0^1 \left(f_1 + (f_2 - f_1)\sqrt{r} \right) dr}{\int_0^1 dr} \right)^2 \quad ; \quad 0 \le r \le 1$$

$$\left\langle p^2{}_{impact} \right\rangle = p^2{}_{max} \cdot \left(f_1 + \frac{2}{3}(f_2 - f_1) \right)^2 = \frac{p^2{}_{max}}{9} \cdot \left(f_1 + 2 f_2 \right)^2 \tag{1.25}$$

$$\left\langle p^2{}_{impact} \right\rangle = \frac{p^2{}_{max}}{9} \cdot \left(f_1^2 + 4 f_2^2 + 4 f_1 f_2 \right) = \frac{1}{9}\left(p_1^2 + 4 p_2^2 + 4 p_1 p_2 \right)$$

Typically we have p1 >> p2, and therefore

$$\left\langle p^2{}_{impact} \right\rangle \approx \frac{4 p^2{}_{max}}{9} f_1^2 = \frac{4}{9} p_1^2 \tag{1.26}$$

Example:

1. For the whole angular rang 0 -180 deg with p_1=0 and p_2=pmax. We find

$$\left\langle p^2{}_{impact} \right\rangle \approx \frac{4 p^2{}_{max}}{9} \,. \tag{1.27}$$

2. For an angular range 130 -180 deg

$$\left\langle p^2{}_{impact} \right\rangle \approx \frac{4 p^2{}_{max}}{9} f_{130}^2 \,. \tag{1.28}$$

Compared to the average impact parameter of eq. (1.27) with $p_{max} \approx$ 1-2Å , the choice of a smaller impact parameter leads to an amplification a of the scattering probability given as

$$a = \frac{\left\langle p_{\max}^2 \right\rangle}{\left\langle p_{impact}^2 \right\rangle} \tag{1.29}$$

The impact parameter is individual for each target element atomic number. In addition, the cross sections for different target elements depend on the square of the target atomic number. In the following some examples are given.

500 keV He on Au: p_{impact} (130°) = 1.04E-3 Å amplification a =2.053E+6
500 keV He on Si: p_{impact} (130°) = 1.85E-4 Å amplification a = 6.486E+7
500 keV He on Au/Si: p_{max} = 1.49 Å
$(Z_{Au}/Z_{Si})^2$ = 31.8418

500 keV He on Au: p_{impact} (90°) = 2.187E-3 Å amplification a= 4.46E+5
500 keV He on Si: p_{impact} (90°) = 3.98E-4 Å amplification a = 1.402E+6
500 keV He on Au/Si: p_{max} = 1.49 Å
$(Z_{Au}/Z_{Si})^2$ = 31.8418

500 keV He on Au: p_{impact} (20°) = 1.23E-2 Å amplification a = 1.4674E+4
500 keV He on Si: p_{impact} (20°) = 2.28E-3 Å amplification a = 4.27E+5
500 keV He on Au/Si: p_{max} = 1.49 Å
$(Z_{Au}/Z_{Si})^2$ = 31.8418

[redacted] Calculation of enforced scattering in IMINTDYN

Enforced scattering is defined by the reduction factor f_{red} (in the program pmax_coeff) , which reduced the impact parameter and increases the scattering angle. The lower and upper values of pmax, pmax1 and pmax2 are calculated based on the selected scattering angle regime (here, scattering at a target element, not the emission angle of a projectile)

Enforced scattering requires one of the four scattering modes

Scattering_mode = „ebs“, „erda“, „cerda“, „ersc“,“nra“

We can distinguish two cases:

a) CERDA coincidence scattering with identical masses M_1 and M_2 and identical atomic numbers Z_1 and Z_2
b) Backscattering with $M_1 << M_2$
c) ERCS with $M_2 \approx 3M_1$ up to $4M_1$

For pure Coulomb scattering, the scattering angle in the lab system is related to the scattering angle in the CM system by

$$\tan \vartheta_{lab} = \frac{M_2 \sin \vartheta_C}{M_1 + M_2 \cos \vartheta_C}. \tag{1.1}$$

The impact parameter is given by

$$p_{impact} = \frac{Z_1 Z_2 e^2}{4\pi\varepsilon_0 E_{kin}} \frac{1}{\tan\left(\vartheta_c / 2\right)} . \tag{1.2}$$

In the case a) CERDA we find the relation

$$\tan \vartheta_{lab} = \frac{\sin \vartheta_C}{1 + \cos \vartheta_C} , \tag{1.3}$$

or

$$\vartheta_{lab} = \frac{1}{2} \vartheta_C , \tag{1.4}$$

With range $0 \le \vartheta_{lab} \le 90°$. In this case we find an impact parameter

$$p_{impact}(CERDA) = \frac{Z_1 Z_2 e^2}{4\pi\varepsilon_0 E_{kin}} \frac{1}{\tan \vartheta_{lab}} . \tag{1.5}$$

The max impact parameter is given by the atomic density n

$$p_{\max} = \frac{1}{\sqrt{\pi}\sqrt[3]{n}} . \tag{1.6}$$

For a given minimum scattering angle, the reduction factor *f_{red}* is

$$f_{red}(CERDA) = \frac{p_{mpact}(CERDA)}{p_{\max}} = \frac{Z_1 Z_2 e^2}{4\pi\varepsilon_0 E_{kin}} \frac{1}{\tan \vartheta_{lab}} \sqrt{\pi}\sqrt[3]{n} \tag{1.7}$$

In the case b) Elastic Backscattering (EBS) we find the relation

$$\tan \vartheta_{lab} \approx \frac{\sin \vartheta_C}{\cos \vartheta_C} = \tan \vartheta_C , \tag{1.8}$$

or

$$\vartheta_{lab} \approx \vartheta_C . \tag{1.9}$$

With range $0 \le \vartheta_{lab} \le 180°$. In this case we find an impact parameter

$$p_{impact}(EBS) \approx \frac{Z_1 Z_2 e^2}{4\pi\varepsilon_0 E_{kin}} \frac{1}{\tan\left(\vartheta_{lab} / 2\right)} . \tag{1.10}$$

The impact parameter for ebs is about a factor 2 too large. Therefore, a correction factor was introduced.

The max impact parameter is given by the atomic density n

$$p_{\max} = \frac{1}{\sqrt{\pi}\sqrt[3]{n}} . \tag{1.11}$$

For a given minimum scattering angle, the reduction factor f_{red} is

$$f_{red}(EBS) = \frac{p_{mpact}(EBS)}{p_{\max}} = \frac{Z_1 Z_2 e^2}{4\pi\varepsilon_0 E_{kin}} \frac{1}{\tan\left(\vartheta_{lab}/2\right)} \sqrt{\pi}\sqrt[3]{n} \tag{1.12}$$

In the case c) ERCS we find the relation

$$\tan\vartheta_{lab} \approx \frac{\sin\vartheta_C}{1/3+\cos\vartheta_C} , \tag{1.13}$$

or

$$\vartheta_{lab} \approx 0.75\cdot\vartheta_C . \tag{1.14}$$

With range $0 \le \vartheta_{lab} \le 90°$. In this case we find an impact parameter

$$p_{impact}(EBS) \approx \frac{Z_1 Z_2 e^2}{4\pi\varepsilon_0 E_{kin}} \frac{1}{\tan\left(\frac{2}{3}\vartheta_{lab}\right)} . \tag{1.15}$$

The max impact parameter is given by the atomic density n

$$p_{\max} = \frac{1}{\sqrt{\pi}\sqrt[3]{n}} . \tag{1.16}$$

For a given minimum scattering angle, the reduction factor f_{red} is

$$f_{red}(EBS) = \frac{p_{mpact}(EBS)}{p_{\max}} = \frac{Z_1 Z_2 e^2}{4\pi\varepsilon_0 E_{kin}} \frac{1}{\tan\left(\frac{2}{3}\vartheta_{lab}\right)} \sqrt{\pi}\sqrt[3]{n} . \tag{1.17}$$

Instead of defining a fixed reduction factor, we define a scattering angle regime and calculate the reduction factor as function of projectile energy using impact parameters determine from the screened potentials used in the simulation.

The regime where enforced scattering can occur depends on the scattering mode (CERDA,EBS, ERCS) and the target thickness. For CERDA it is typically the target thickness. For EBS the program first calculates the residual energy of e.g. backscattered ions and from that the corresponding target depth of scattering. Enforced scattering then occurs only up to this target depth.

[redacted] Simulation of nuclear reactions

IMINTYN can simulate nuclear reactions in a way that the projectile vanished in a reaction and a new recoil species is generated with the proper energy and angular distribution. These “recoils” are then collected as transmitted or backemitted particles. The4 procedure is therefore similar to and ERDA simulation.

To activate NRA simulations one used the command

Scattering mode=”nra”

NRA target atom species and NRA product atom species can be defined by

```
nra_target =
nra_product =
```

Up to ncpm target and corresponding product elements for NRA simulations can be defined.

Example: mra_target = “Li7”,”Li6” ; nra_product=”He4”,”He3”.

The relative reaction cross sections for different projectile energies as well as the different target concentrations with depth are stored in the data output file as weight factors.

The cross section data files for non-Rutherford scattering as well as Nuclear Reaction Analysis are taken from the ion beam analysis nuclear data library IBANDL files [16].

`file_r33 =` An IBANDL non-RBS cross section files up to 3 files per target species can be used, total 3*ncpm files

`file_r33_nra =` An IBANDL NRA cross section files up to 3 files per target species can be used, total 3*ncpm files

These files are collected in the directory ../tables/nonrbs

4.10 Simulation of dimer sputtering

The SDTrimSP program from MPI Plasmaphysics uses the surface binding energy as fit parameter. Sputter yields for carbon compounds use E_{SBE}=4.5 eV instead of the sublimation energy of 7.4 eV. Sputtering of oxide compounds is often adjusted by setting E_{SBE} =1 eV for oxygen instead of the sublimation energy of 2.58 eV. One possible origin for the experimental higher sputter yields can be dimer sputtering. Dimer sputtering was investigated by Oyarabal et al. in detail for noble gas ion irradiation of carbon and significant contribution of dimer sputtering was found [18]. In this chapter we try to implement dimer sputtering into IMINTDYN so that only tabulated thermodynamic sublimation and formation enthalpies are used in the simulations.

The table “table-compounds-dimer” contains data on binary compounds used for sputtering of dimers. It contains the formation enthalpy of the stoichiometric compound A_xB_y and also the formation enthalpies of the dimers AA, BB, and AB. In some cases the dimer is the thermodynamic reference level and has a formation enthalpy of zero (example: O_2 or N_2

molecule or graphite solid). The formation enthalpies are then given with respect to thermodynamic stable compound or species. Examples are

$$Si(solid) + O_2\,(gas) \rightarrow SiO_2\ solid$$

$$C\ (graphite, solid) + O_2\,(gas) \rightarrow CO_2$$

In IMINTDYN the best approach is to use elemental sublimation enthalpies to describe sputtering of monomers. It seems that the formation of a compound does not play a role. A compound is actually not disintegrated but only a single atom is removed into vacuum. The elemental sublimation energy values in unit eV/atom are tabulated in the table “table-elements.txt”. Formation enthalpies of binary compounds are tabulated in “table-compounds.txt”, however, the use formation enthalpies typically leads to strongly reduced sputtering yields.

A new table “table-compounds-dimer.txt” contains additional formation enthalpies for the formation of dimers. Most values are taken from the NIST Chemistry Web book [17]

[redacted] Formation of dimers from elemental targets

4.10.1.1 Carbon dimers

As a first example we consider sputtering of graphite. The elemental sublimation enthalpy for C is 716.68 kj/mol or 7.4 eV/atom. This is the energy required to remove one C atom from the solid into vacuum. The conversion from kJ/mol into eV/atom is done using

$$f = \frac{1000}{e \cdot N_A} = \frac{1000}{1.602 \cdot 10^{-19} C \cdot 6.023 \cdot 10^{23}\ atom/mol} = 0.0103644 \frac{mol}{C \cdot atom} \qquad (1.18)$$

For a C2-dimer we find 837.74 kJ/mol or 8.6826 eV/dimer.

Removal of two C atoms into vacuum requires an energy of 2·7.4 eV = 14.8 eV. Removal of a C2-dimer into vacuum requires only 8.6826 eV/dimer. The dissociation energy of a C2 dimer is therefore E_{dis} = 6.117366 eV

In a sputtering process we now have the choice to remove a C monomer consuming 7.4 eV or removing a C-dimer consuming 8.6826 eV. In the latter case, the sputter yield would be a factor two larger. However, the kinetic energy of the projectile leaving the surface must exceed at least the energy value for dimer formation, i.e. 8.62826 eV

We now have to consider the probabilities for both processes. We use a Boltzmann factor with the elemental sublimation enthalpy $E_{C\text{-}monomer}$ =7.4eV as reference value. We then get

$$\tilde{p}_{monomer} = \exp\left(-\frac{E_{C-monomer}}{E_{C-monomer}}\right) = \exp(-1) = 0.36788 \qquad (1.19)$$

$$\tilde{p}_{dimer} = \gamma_{dimer} \exp\left(-\frac{E_{C-dimer}}{E_{C-monomer}}\right) = \gamma_{dimer} \exp\left(-\frac{8.6826}{7.4}\right) = 0.309337 \qquad (1.20)$$

The factor γ_{dimer} is an empirical factor describing the dimer formation efficiency. Apart from thermodynamic considerations this efficiency may be related to bond structure arguments, so that for some compounds it is rather unlikely to release dimers or trimers. For our discussion here we use the default value

$$\gamma_{dimer} = 1. \tag{1.21}$$

The partition function is used for normalization:

$$Z = \sum_i \gamma_i \exp\left(-\frac{E_i}{E_{C-monomer}} \right) \tag{1.22}$$

For the above case with two species we find

$$Z_{graphite} = 0.677217 \tag{1.23}$$

The final probabilities for monomer and dimer sputtering are then given as:

$$p_{monomer} = \frac{1}{Z} \exp\left(-\frac{E_{C-monomer}}{E_{C-monomer}} \right) = 54.32\% \tag{1.24}$$

$$p_{\dim er} = \frac{1}{Z} \exp\left(-\frac{E_{C-\dim er}}{E_{C-monomer}} \right) = 45.68\% \tag{1.25}$$

Compared to only a monomer sputter yield Y_m , the sputter yield then increases to

$$Y = Y_m \cdot \left(p_{monomer} + 2 p_{\dim er} \right) = Y_m \cdot (1 \cdot 0.5432 + 2 \cdot 0.4568) = Y_m \cdot 1.4568 \tag{1.26}$$

Because of dimer sputtering, the sputter yield will be a factor of 1.46 higher!

According to the NIST Chemistry Web book there should be a probability for emission of C-trimers and C_4 clusters with formation enthalpies of 820.06 kJ/mol and 970.69 kJ/mol, respectively. The question is if such clusters can be generated in a single atom collision and if the projectile energy is sufficiently high.

$$\tilde{p}_{trimer} = \gamma_{trimer} \exp\left(-\frac{E_{C-trimer}}{E_{C-monomer}} \right) = \exp\left(-\frac{8.499}{7.4} \right) = 0.3171 \tag{1.27}$$

$$\tilde{p}_{tetramer} = \gamma_{tetramer} \exp\left(-\frac{E_{C-tetramer}}{E_{C-monomer}} \right) = \exp\left(-\frac{10.0606}{7.4} \right) = 0.25677 \tag{1.28}$$

Partition function:

$$Z_{graphite} = 1.2511 \tag{1.29}$$

The final probabilities for monomer, dimer, trimer and tetramer sputtering are then given as:

$$p_{monomer} = \frac{1}{Z}\exp\left(-\frac{E_{C-monomer}}{E_{C-monomer}}\right) = 29.40\% \tag{1.30}$$

$$p_{\dim er} = \frac{1}{Z}\exp\left(-\frac{E_{C-\dim er}}{E_{C-monomer}}\right) = 26.73\% \tag{1.31}$$

$$p_{\dim er} = \frac{1}{Z}\exp\left(-\frac{E_{C-trimer}}{E_{C-monomer}}\right) = 25.35\% \tag{1.32}$$

$$p_{\dim er} = \frac{1}{Z}\exp\left(-\frac{E_{C-tetramer}}{E_{C-monomer}}\right) = 20.52\% \tag{1.33}$$

Compared to only a monomer sputter yield Y_m , the sputter yield then increases to

$$Y = Y_m \cdot \left(p_{monomer} + 2p_{\dim er} + 3p_{rimer} + 4p_{tetramer}\right) = Y_m \cdot (1 \cdot 0.294 + 2 \cdot 0.2673 + 3 \cdot 0.2535 + 4 \cdot 0.2052) = Y_m \cdot 2.4099 \tag{1.34}$$

Because of oligomer sputtering, the sputter yield will be a factor of 2.41 higher!

4.10.1.2 Aluminum dimers

Tabulated enthalpy values for Al_2: 5.0455 eV

Tabulated elemental value for Al: 3.417 eV

Probablities for sputtering using Boltzmann factors:

Al monomer: $\exp(-\Delta_f H_{Al} / -\Delta_f H_{Al}) = 0.36$ Al dimer: $\exp(-\frac{\Delta_f H_{Al2}}{\Delta_f H_{Al}}) = 0.228$;

Partition function: Z = 0.36 + 0.228 = 0.588
Sputter fractions: Al-monomer 0.36/Z = 61 % Al-dimer: 0.228/Z = 39 %
Effective rel. sputter yield: 1.39

4.10.1.3 Silicon dimers

Tabulated enthalpy value for Si_2: 6.1142 eV

Tabulated elemental value for Si: 4.67 eV

Probablities for sputtering using Boltzmann factors:

Si monomer: $\exp(-\Delta_f H_{Si} / -\Delta_f H_{Si}) = 0.36$

Si dimer: $\exp(-\frac{2\Delta_f H_{Si} + \Delta_f H_{Si2}}{\Delta_f H_{Si}}) = 0.27$;

Partition function: Z = 0.36 + 0.27 = 0.63
Sputter fractions: Si-monomer 0.36/Z = 57 % Si-dimer: 0.036/Z = 43 %
Effective rel. sputter yield: 1.43

4.10.1.4 Cu dimers and others

Tabulated enthalpy value for Cu_2: 485.34 kJ/mol	**for Cu: 337.4 kJ/mol**
Tabulated enthalpy value for Mg_2: 287.63 kJ/mol	for Mg: 147.1 kJ/mol
Tabulated enthalpy value for Pb_2: 332.63 kJ/mol	for Pb: 195.2 kJ/mol
Tabulated enthalpy value for B_2: 829.69 kJ/mol	for B: 565 kJ/mol
Tabulated enthalpy value for Li_2: 215.9 kJ/mol	for Li: 159.3 kJ/mol
Tabulated enthalpy value for P_2: 144.0 kJ/mol	for P: 316.4 kJ/mol
Tabulated enthalpy value for S_2: 128.6 kJ/mol	for S: 276.98 kJ/mol
Tabulated enthalpy value for K_2: 123.68 kJ/mol	for K: 89 kJ/mol

Probablities for sputtering using Boltzmann factors:

Cu monomer: $\exp(-\Delta_f H_{Cu} / -\Delta_f H_{Cu}) = 0.36$ Cu dimer: $\exp(-\frac{\Delta_f H_{Cu2}}{\Delta_f H_{Cu}}) = 0.237$;

Partition function: Z = 0.36 + 0.237 = 0.597
Sputter fractions: Si-monomer 0.36/Z = 60 % Si-dimer: 0.036/Z = 40 %
Effective rel. sputter yield: 1.40

4.10.1.5 Nickel dimers and others

Tabulated enthalpy value for Ni_2: none

Tabulated elemental value for Ni: 430.12 kJ/mol or 4.458 eV

No dimer sputtering

Also no dimer sputtering for Cr, Co, Fe, Mn, Ge, Mo, Nb, Sn, Pd, Ag, Ta, W, Au

[redacted]Formation of dimers from compound targets

4.10.2.1 Silicondioxide

Sputtering of SiO_2 can result in emission of Si, O, SiO Si_2 or O_2 species.

O_2 ist the thermodynamic stable dimer and the enthalpy to form O_2 from two O atoms is $2 \cdot H_{subl} = -2 \cdot 2.58\ \text{eV/atom} = -2 \cdot 249.17\ \text{kJ/mol O} = -5.16\ \text{eV/molecule}$. The minus sign indicates that the reaction is exothermic.

The formation enthalpy for SiO is

$$2\ Si(solid) + O_2(gas) \rightarrow 2\ SiO \quad -100.42 kJ/mol; -1.04 eV/molecule \qquad (1.35)$$

To dissociate an O_2 molecule require 5.16 eV/molecule or 2.58 eV/atom
The formation energy of gas phase Si atoms from Si solid is 450 kJ/mol or 4.67 eV/atom.
The formation energy of SiO molecules from Si solid + O_2 is -100.42 kJ/mol SiO
or -1.04 eV/molecule.
To evaporate a SiO molecule is energetically favourable an releases -1.04 eV/molecule
The formation energy of gas phase Si_2 molecules from Si solid is 589.94 kJ/mol Si_2
or 6.1143 eV/molecule.
To remove two Si atoms from a target using elemental enthalpies requires 2·4.67 eV = 9.34 eV
To dissociate a Si_2 molecule requires 9.34 - 6.1142 eV = 3.228 eV/molecule.
The formation energy of O_2 molecules from gas phase O atoms is -498.34 kJ/mol Si_2
or -5.16 eV/molecule.
To remove two O atoms from a target using elemental enthalpies requires 2·2.58 eV = 5.16 eV
To remove an O_2 molecule requires 0 eV
The formation energy of SiO_2 quartz Si solid + O_2 is -986 kJ/mol or -10.22 eV or -3.4 eV/atom
To remove one Si and one O atom from a target using elemental enthalpies requires
2.58 +4.67 eV = 7.45 eV

Tabulated enthalpy values for SiO_2: eV SiO: - 1.04 eV Si_2: 6.1142 eV O_2: 0 eV

Tabulated elemental value for Si: 4.67 eV for O: 2.58 eV

Probabilities for sputtering using Boltzmann factors:

Here we have to consider a probability of $\frac{1}{3}$ to find a Si atom as 2nd partner and of $\frac{2}{3}$ to find an O atom as 2nd partner

O monomer: $\exp(-\Delta_f H_O / -\Delta_f H_O) = 0.36$ O dimer: $\frac{2}{3}\exp(-\frac{\Delta_f H_{O2}}{\Delta_f H_O}) = \frac{2}{3}$; $\frac{2}{3} = \frac{n}{n+m}$

Si monomer: $\exp(-\Delta_f H_{Si} / -\Delta_f H_{Si}) = 0.36$

Si dimer: $\frac{1}{3}\exp(-\frac{\Delta_f H_{Si2}}{\Delta_f H_{Si}}) = \frac{1}{3}\exp(-1.309) = 0.09$

SiO dimer from O projectile: $\frac{1}{3}\exp(-\frac{\Delta_f H_{SiO}}{\Delta_f H_O}) = \frac{1}{3}\exp(-\frac{-1.04}{2.58}) = 0.4988$

SiO dimer from Si projectile: $\frac{2}{3}\exp(-\frac{\Delta_f H_{SiO}}{\Delta_f H_{Si}}) = \frac{2}{3}\exp(-\frac{-1.04}{4.67}) = 0.833$

<u>Sputtering with O as projectile:</u>

Partition function: Z = 0.36 + 2/3 + 0.4988 = 1.5255
Sputter fractions: O-monomer 0.36/Z = 23.6 %
O-dimer: 2/3Z = 43.7 % SiO-dimer: 0.4988/Z = 32.7 %
Effective rel. sputter yield enhancement: 1.764

<u>Sputtering with Si as projectile:</u>
Partition function: Z = 0.36 + 0.09 + 0.833 = 1.283
Sputter fractions: Si-monomer 0.36/Z = 28 %
O-dimer: 0.09/Z = 7 % SiO-dimer: 0.833/Z = 65 %
Effective rel. sputter yield enhancement: 1.72

4.10.2.2 Tantalum oxide Ta_2O_5

Sputtering of Ta_2O_5 can result in emission of Ta, O, TaO, Ta_2 or O_2 species.

O_2 ist the thermodynamic stable dimer and the enthalpy to form O_2 from two O atoms is $2 \cdot H_{subl} = -2 \cdot 2.58\,\text{eV/atom} = -2 \cdot 249.17\,\text{kJ/mol O} = -5.16\,\text{eV/molecule}$. The minus sign indicates that the reaction is exothermic.

The formation enthalpy for TaO is

$$2\,Ta(solid) + O_2(gas) \rightarrow 2\,TaO \quad 142.46 kJ/mol; +1.4756 eV/molecule \qquad (1.36)$$

To dissociate an O_2 molecule require 5.16 eV/molecule or 2.58 eV/atom
The formation energy of gas phase Ta atoms from Ta solid is 782 kJ/mol or 8.10 eV/atom.
The formation energy of TaO molecules from Ta solid + O_2 is +142.46 kJ/mol TaO or +1.4756 eV/molecule.
The formation energy of gas phase Ta_2 molecules from Ta solid: no data available.
To remove two Ta atoms from a target using elemental enthalpies requires 2·8.1 eV = 16.2 eV
The formation energy of O_2 molecules from gas phase O atoms is -498.34 kJ/mol Si_2 or -5.16 eV/molecule.
To remove two O atoms from a target using elemental enthalpies requires 2·2.58 eV = 5.16 eV
To remove an O_2 molecule requires 0 eV

To remove one Ta and one O atom from a target using elemental enthalpies requires 8.10 + 2.58 eV = 10.68 eV

Tabulated enthalpy values for TaO: 1.4756 eV Ta_2: ----- O_2: 0 eV

Tabulated elemental value for Ta: 8.1 eV for O: 2.58 eV

Probabilities for sputtering using Boltzmann factors:

Here we have to consider a probability of $\frac{2}{7}$ to find a Ta atom as 2nd partner and of $\frac{5}{7}$ to find an O atom as 2nd partner

O monomer: $\exp(-\Delta_f H_O / -\Delta_f H_O) = 0.36$ O dimer: $\frac{5}{7}\exp(-\frac{\Delta_f H_{O2}}{\Delta_f H_O}) = \frac{5}{7}$; $\frac{2}{3} = \frac{n}{n+m}$

Ta monomer: $\exp(-\Delta_f H_{Ta} / -\Delta_f H_{Ta}) = 0.36$ Ta dimer: $\frac{2}{7}\exp(-\frac{\Delta_f H_{Ta2}}{\Delta_f H_{Ta}}) = 0$

TaO dimer from Ta projectile: $\frac{5}{7}\exp(-\frac{\Delta_f H_{TaO}}{\Delta_f H_{Ta}}) = \frac{5}{7}\exp(-\frac{1.4756}{8.1}) = 0.5953$

TaO dimer from O projectile: $\frac{2}{7}\exp(-\frac{\Delta_f H_{TaO}}{\Delta_f H_O}) = \frac{2}{7}\exp(-\frac{1.4756}{2.58}) = 0.161$

<u>Sputtering with O as projectile:</u>

Partition function: Z = 0.36 + 5/7 + 0.161 = 1.2353
Sputter fractions: O-monomer 0.36/Z = 29.2 % O-dimer: 5/7Z = 57.8 % TaO-dimer: 0.4165/Z = 13 %
Effective rel. sputter yield enhancement: 1.708

<u>Sputtering with Ta as projectile:</u>
Partition function: Z = 0.36 + 0.00 + 0.5953 = 0.9553
Sputter fractions: Ta-monomer 0.36/Z = 37.7 % Ta-dimer: 0.00/Z = 0 % TaO-dimer: 0.833/Z = 62.3 %
Effective rel. sputter yield enhancement: 1.623

4.10.2.3 Siliconnitride Si_3N_4

Sputtering of Si_3N_4 can result in emission of Si, N, SiN Si_2 or N_2 species.

N_2 ist the thermodynamic stable dimer and the enthalpy to form N_2 from two N atoms is $2\cdot H_{subl} = -2\cdot 4.9\,\text{eV/atom} = -2\cdot 472.68\,\text{kJ/mol N} = -9.8\,\text{eV/molecule}$. The minus sign indicates that the reaction is exothermic.

The formation enthalpy for SiN is

$$2\,Si(solid) + N_2(gas) \rightarrow 2\,SiN \quad +372.38 kJ/mol; +3.68 eV/molecule \tag{1.37}$$

To dissociate an N_2 molecule requires 9.8 eV/molecule or 4.9 eV/atom
The formation energy of gas phase Si atoms from Si solid is 450 kJ/mol or 4.67 eV/atom.
The formation energy of SiN molecules from Si solid + N_2 is +372.38 kJ/mol SiN
or +3.68 eV/molecule.
The formation energy of gas phase Si_2 molecules from Si solid is 589.94 kJ/mol Si_2
or 6.1143 eV/molecule.
To remove two Si atoms from a target using elemental enthalpies requires
2·4.67 eV = 9.34 eV
To dissociate a Si_2 molecule requires 9.34 - 6.1142 eV = 3.228 eV/molecule.
The formation energy of N_2 molecules from gas phase N atoms is -472.68 kJ/mol Si_2
or -9.8 eV/molecule.
To remove two N atoms from a target using elemental enthalpies requires 2·4.9 eV = 9.8 eV
To remove an N_2 molecule requires 0 eV
To remove one Si and one N atom from a target using elemental enthalpies requires
4.67+ 4.9 eV = 9.57 eV

Tabulated enthalpy values SiN: 3.68 eV Si$_2$: 6.1142 eV N$_2$: 0 eV

Tabulated elemental value for Si: 4.67 eV for N: 4.9 eV

Probabilities for sputtering using Boltzmann factors:

Here we have to consider a probability of $\frac{3}{7}$ to find a Si atom as 2nd partner and of $\frac{4}{7}$ to find a N atom as 2nd partner

N monomer: $\exp(-\Delta_f H_N / -\Delta_f H_N) = 0.36$ N dimer: $\frac{4}{7}\exp(-\frac{\Delta_f H_{N2}}{\Delta_f H_N}) = \frac{4}{7}$;

Si monomer: $\exp(-\Delta_f H_{Si} / -\Delta_f H_{Si}) = 0.36$

Si dimer: $\frac{3}{7}\exp(-\frac{\Delta_f H_{Si2}}{\Delta_f H_{Si}}) = \frac{3}{7}\exp(-1.309) = 0.115$

SiN dimer from N projectile: $\frac{3}{7}\exp(-\frac{\Delta_f H_{SiN}}{\Delta_f H_N}) = \frac{3}{7}\exp(-\frac{3.68}{4.9}) = 0.2022$

SiN dimer from Si projectile: $\frac{4}{7}\exp(-\frac{\Delta_f H_{SiN}}{\Delta_f H_{Si}}) = \frac{4}{7}\exp(-\frac{3.68}{4.67}) = 0.2599$

Sputtering with N as projectile:

Partition function: Z = 0.36 + 4/7 + 0.2022 = 1.1336
Sputter fractions: N-monomer 0.36/Z = 31.8 % N-dimer: 4/7Z = 50.4 % SiN-dimer: 0.2022/Z = 17.8 %
Effective rel. sputter yield enhancement: 1.683

Sputtering with Si as projectile:
Partition function: Z = 0.36 + 0.115 + 0.2599 = 0.7349
Sputter fractions: Si-monomer 0.36/Z = 49 % N-dimer: 0.09/Z = 15.6 % SiN-dimer: 0.2599/Z = 35.4 %
Effective rel. sputter yield enhancement: 1.51

[redacted] Simulation of dimer sputtering in IMINTDYN

This part is included in the subroutines "RECOIL.F90" and "TABLEREAD.F90"

Prerequistes are that

1. A binary or multi-component target is specified
2. The flag dimer_sputtering=.true., is set
3. A dimer_compound is specified.
4. A dimer formation efficiency γ_{dimer} is defined

The program reads data from the table-dimers.txt and sets the formation enthalpies for dimer formation for a dimer with atomic numbers Z1 and Z2. Three enthalpies for Z1Z1, Z2,Z2 and Z1Z2 are extracted.

Dimer formation is an inelastic collision process, which only obeys momentum conservation. Compared to the initial kinetic energy of a monomer (Z1), the kinetic energy of the dimer Z1-Z2 is reduced by a factor

$$f_r = \frac{\text{atomic mass}(Z2)}{\text{atomic mass(Z1)+atomic mass(Z2)}} \tag{1.38}$$

This reduced kinetic energy is in addition modified by the dimer formation energy, which also can be negative in case of an exothermic process. The dimer energy is then given as

$$E_{\text{dimer}} = f_R E_{kin} - E_{formation} \tag{1.39}$$

For sputtering of the monomer, we compare the normal kinetic energy component with the surface binding energy of monomers, or zero in case of the bulk binding energy model.

For sputtering of a dimer we use the formation energy of the respective dimers Z_1Z_1, Z_2Z_2 or Z_1Z_2. The normal component of the kinetic energy for the dimer is then compared with this formation energy in case of the surface binding energy model.

In the subroutine “recoil2 we first select a projectile with Z_1 and kinetic energy. If this projectile is being sputtered, we select a second target atom based on the stoichiometry in the first three target layers using random numbers. This 2nd target atoms (lets say with Z_2) then forms a dimer Z_1Z_2 with reduced energy but with the same velocity direction as the initial monomer.

To calculate the probability for monomer od dimer sputtering we first calculate a partition function using the monomer formation enthalpy as reference. The partition function Z is

$$Z = \exp\left(-\frac{enthalpy(Z_1)}{enthalpy(Z_1)}\right) + \gamma_{\mathrm{dim}er} \exp\left(\frac{enthalpy(Z_1Z_2)}{enthalpy(Z_1)}\right) \tag{1.40}$$

The probablity for monomer sputtering is the given as

$$p_{monomer} = \frac{1}{Z}\exp\left(-\frac{enthalpy(z1)}{enthalpy(z1)}\right) \tag{1.41}$$

The probablity for dimer sputtering is the given as

$$p_{\mathrm{dim}er} = \frac{1}{Z}\gamma_{dimer}\exp\left(-\frac{enthalpy(Z_1Z_2)}{enthalpy(Z_1)}\right) \tag{1.42}$$

We always have

$$p_{monomer} + p_{\mathrm{dim}er} = 1 \tag{1.43}$$

In case that monomer- or dimer-sputtering is energetically possible, we then calculate the sputter yield by using counters which are increased by a value $p_{monomer}$ and p_{dimer}.

As a result, we get the conventional monomer sputtering yield for components Z_1 and Z_2 if dimer sputtering is neglected, but also the partial sputtering yields for Monomer(Z_1), Monomer(Z_2), dimer(Z_1Z_1), Dimer Z_1Z_2) and Dimer(Z_2Z_2) as well as Trimer ($Z_2Z_2Z_2$).

For many metal atoms, a dimer formation is unlikely. In these cases, an unusual formation enthalpy of 9999 eV/dimer is read from the table, so that the corresponding term in the partition function becomes zero.

4.11 Target layer structure definition

Typically, a target its defined by its target thickness `target_thickness`, its elemental composition and the number `nqx` of planar target slices. The minimum requirement to define a target is for example for a SiO2 target

```
symbol = "Ar","Si","O"          ! names of projectile and target elements
```

```
target_thickness =  1000E+0         ! target depth (Angstroems)
nqx     =   500E+0            ! number of target intervals
atomic_fraction = 0.,1., 2.   ! initial concentration in the target
                              ! normalization done by the program
max_atomic_fraction = 0.200, 1.000, 1.000 ! max concentration
```

The first value `atomic_fraction = 0`. May define a projectile, which is initially not present in the target

The values `max_atomic_fraction` define the maximum allowed concentration above which outdiffusion sets in. In addition, one defines the beam and target components as

```
beam_fraction = 1.000, 0.000, 0.000 ! fraction of incident beam
```

and the electrical charge of the incoming ions as

```
charge = 1. , 0.0,0.00        ! electrical charge
```

Depending on the ion irradiation process a target may grow by deposition or erode by sputter erosion. The latter process may eventually lead to a complete loss of the target. To circumvent this, we have added the new option `add_bulk_layers=.true.` ,which stabilizes the number of target layer at its initial value. If nqx drops below this value because of erosion processes, a layer with composition and thickness of the last layer is added as new last layer. In this way the target substrate becomes infinite thick and the number of layer stabilizes at its initial value.

```
add_bulk_layers=.true.  automatically ad a last layer if nqx

                                drops below its initial value

add_bulk_layers=.false. disable this option
```

A more complex layered target structure can be specified in a target layer definition file with file extension ".def". In this file the composition of target elements is specified as function of the layer thickness. The target elements are not specified, so that the same layer definition file can be used for different target and projectiles species. These have to be specified by their atomic symbols, e.g.

```
symbol = "Ar,"Si,"O"
```

As an example, we consider a multilayer where each layer has a thickness of Å. We define a structure for three species 1,2,3. The first two lines of the .def file are text lines.

We start with alternating layers of species 2 and 3 and a thick substrate layer. The composition varies between 0.33/0.67 (e.g. SiO2) and 1.0/0. (e.g. Si) and the layers vary between 20 and 30. At the end we have 50 layers of 30Å each

We have an alternating composition of 100% for element species 2 and 3. Element species 1 is supposed to be the incident ion and has initial composition 0. In total there are 500 layers of 3Å each distributed over the target thickness target_thickness= 500*3 Å = 1500 Å.

-1 in the last line indicates the end of the layered structure. If this is replaced by 0, then the previous layered structure is repeated until the maximum of nqxm layers are filled.

Based on the input, the new target thickness will be `target_thickness=1500` with the sum `nqx=500` of the specified layers is calculated.

```
number n    thick-   composition of target(1:num_species);
layers     ness [A]    qu_1...qu_num_species; n<0:end, n=0:copy until nqxm
      20       3.00     0.00  0.33  0.67   ! SiO2 60 Å
      30       3.00     0.00  1.00  0.00   ! Si 90 Å
      20       3.00     0.00  0.33  0.67   ! SiO2 60 Å
      30       3.00     0.00  1.00  0.00   ! Si 90 Å
      20       3.00     0.00  0.33  0.67   ! SiO2 60 Å
      30       3.00     0.00  1.00  0.00   ! Si 90 Å
      350      3.00     0.00  0.33  0.67   ! SiO2 1050 Å
      -1
```

This layer will get a filename such as “Si-SiO2-multilayer.def”.

If the last line has a value 0, then the last layer is copies until the maximum number of layers is reached

In the input script file we then specify:

```
layer_input =.true.
file_layerinp = 'Si-SiO2-multilayer.def'

In the bash.sh script file we then specify the support file
SUPPORT=(Si-SiO2-multilayer)
```

4.12 Energy input files

A distribution of projectile energies can be defined in a file file_energyinp=’energy.def’. This file may be place in a directory defined by dir_energyinp =’./’ .This file contains two header lines of ASCII data followed by up to 200 lines of data pair for energy and probability, separated by space or tab. The probabilities do not need to be normalized. Normalization is done by the program.

```
file_energyinp =’energy.def’  default
dir_energyinp =’./’     default
```

Example: energy distribution with maximum at 2400 eV

```
template for energy distribution input (may be generated with ORIGIN)
energy [eV] (0.1 eV<E<1E9 eV), probability (not normalized), up to 200 lines
50.   1.
100.  2.
120.  5.
150.  7.
400. 12.
500. 17.
800.  22.
1000. 27.
1200. 35.
```

```
1500. 54.
1800. 76.
2400. 81.
2500. 74.
2800. 52.
2900. 34.
3000. 26.
3100. 15.
3400. 12.
3500. 5.
3600. 2.
3700. 1.
```

4.13 Angular input files

A distribution of projectile incidence angles can be defined in a file file_angleinp='angle.def'. This file may be place in a directory defined by `dir_angleinp ='./'`. This file contains two header lines of ASCII data followed by up to 180 lines of data pair for angle and probability, separated by space or tab. The probabilities do not need to be normalized. Normalization is done by the program.

```
file_angleinp='angle.def'     default
dir_angleinp ='./'      default
```

Example: angular distribution with maximum at 45°.

```
template for angular distribution as input (may be generated with ORIGIN)
angle[deg] (0 < alpha < 90deg), probability(not normalized),  up to 180 lines

   28  0
   29  0
   30  1
   31  1
   32  3
   33  5
   34  8
   35  13
   36  19
   37  27
   38  37
   39  48
   40  60
   41  72
   42  83
   43  92
   44  98
   45  100
   46  98
   47  92
   48  83
   49  72
   50  60
   51  48
   52  37
   53  27
```

```
54  19
55  13
56  8
57  5
58  3
59  1
60  1
61  0
62  0
63  0
64  0
65  0
```

4.14 Table of scattering angle versus impact parameter

Such a table can be created, if the script file contains a command

```
impactflag=.true.
```

The program then creates a table of scattering angles for all pairs of defined atomic species and for the interaction potential specified as function of the impact parameter between 10-6 Å and 10 Å. This table allows an evaluation of scattering angles for different interaction potentials and an evaluation of impact parameters needed for large angle scattering in the enforced scattering mode.

4.15 Tables of electronic stopping cross section

Such a table can be created, if the script file contains a command

```
elosstables=.true.
```

The program then creates a table of electronic stopping cross sections for all pairs of defined atomic species and for the stopping model specified as function of the projectile energy up to 2 GeV. This table allows an evaluation of stopping cross section for different stopping models.

4.16 Debugging mode

IMINTDYN provides various debugging modes, which create additional terminal output to evaluate possible problems occuring during simulations and to help debugging program errors. To minimize the terminal output, debugging can be restricted to different part of the program by defining a debugging level. This is specified in the input script file as

```
Debugging = 0 ! 0,...,9 set debugging flag to print output infos
```

0: debugging is off
1: output from subroutines imintdyn.F90 and histories.F90
2: only some output from subroutine histories.F90
3: only output from subroutines imintdyn.F90 and projectile.F90
4: only output from subroutines imintdyn.F90 and recoil.F90

5: only debugging of subroutines set_sublenergy in sub.F90
6: only debugging of subroutines set_temp_cascade in sub.F90
7: only debugging of subroutine free_path in sub.F90
8: only debugging for gsum_fluence.F90 and gsum_fluence2.F90

4.17 Calculation of crater function moments

IMINTDYN allows the calculation of crater function moments up to 6th order. These moments include erosion crater functions, redistribution crater functions and implantation crater functions. The moments are calculated in the subroutines "projectile" and "recoil" and a further analysis of the collision data output is no longer required. In the input script file one only needs to set the command

```
        craterflag=.true.    ! create crater function data file
```

Then the simulation generates the following output files "balancefile.dat", "craterfunction.dat", "cratermoments.dat" and "crater-moments-implantation.dat". These files can be analyzed with an ORIGIN project to calculate all types of curvature coefficients and other data to predict ion induced surface pattern formation.

The calculation of crater function moments also requires the calculation of surface curvature dependent sputter yields. For this purpose one can define a weak surface curvature using the input script commands

```
Kij = 0.00, 0., 0.  ! parabolic surface curvature x_surface(
                    ! y,z) = 1/2 K11 y² + K12 xy + 1/2 K22 z²
x0 = 100.,100.,100. ! origin of impact can be set to positive
                    ! x0 values into the target
```

The program then generates a virtual surface at a depth of 100 Å. The impact point for incident projectiles is set to x_0. The target volume up to the virtual surface is then automatically filled with target atoms. Atoms crossing the virtual surface are considered as sputtered. The curvature should be small, typically curvatures $K_{ij} \leq 0.02$, and x_0 sufficiently large, so that for larger x,y, values the virtual surface is always below the real surface at x0=0. Furthermore, care must be taken in a dynamic simulation which still assumes flat target slices.

4.18 Non-Rutherford cross sections

IMINTDYN can handle non-Rutherford cross sections taken from the IBANDL (Ion Beam Analysis Nuclear Data Library) data base [16]. The program can read cross section data in the so-called R33 format by using the ioput script commands

```
    non_rutherford = .true.         ! use Rutherford cross sections
    file_r33 = "h1pp0$4_2.r33", ...! IBANDL non.RBS cross section file
```

IMINTDYN reads the cross section data files and recalculates the cross sections in units "ratio-to-Rutherford" in 1000 steps between 0 eV and the maximum projectile energy. One can specify up to three files per target species and each file may cover different energy regimes. Overlapping cross section data are overwritten by the 2nd and 3d cross section data file. For missing cross sections a “ratio-to-Rutherford” = 1 is used.

The statistical weight for scattering is added as a new column in the data output files for backscattered, back-reflected and transmitted projectiles and recoils.

The figure shows an example of non-Rutherford cross sections for MeV H ions scattering at N target atoms for a scattering angle of 170°. If one would use Rutherford cross sections, then the RBS spectrum for 2.5 MeV H backscattering at Si3N4 would look like Figure..... Backscattering from Si is the dominating part in the spectrum. Including the non-Rutherford cross section for N we obtain the spectrum shown in figure... Now the N signal is dominant and furthermore shows a peak due to the resonance around 2.3 MeV.

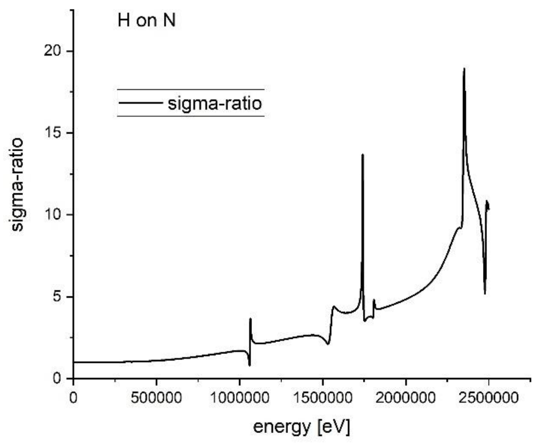

Figure: ratio-to -Rutherford normalized cross section for H projectiles scattered at N target atoms for a scattering angle of 170°.

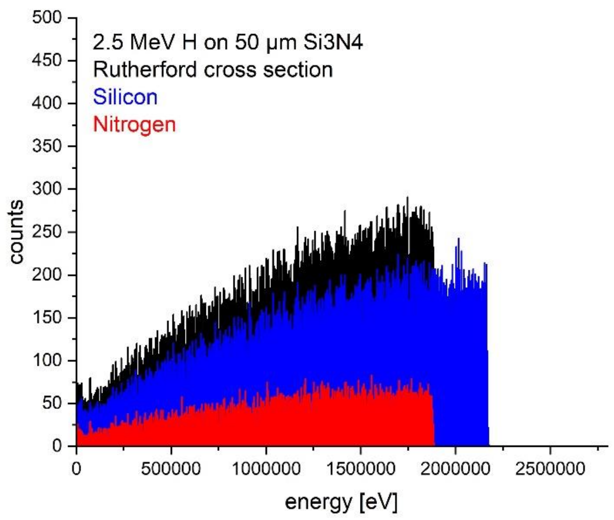

Figure: RBS spectrum for 2.5MeV H backscattering at Si3N4 at 170° for Rutherford cross sections.

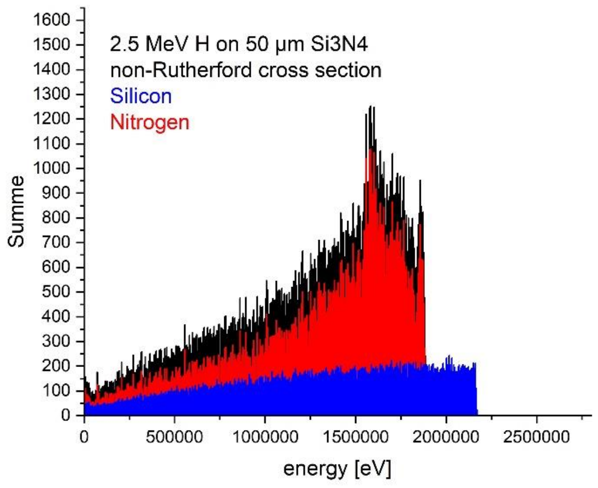

Figure: RBS spectrum for 2.5MeV H backscattering at Si3N4 at 170° for non-Rutherford cross section for N target atoms.

4.19 Time stamp tracking in IMINTDYN

Time tracking is done for all projectiles and recoils starting with time zero at projectile impact. As a basic time step Δt IMINTDYN uses the time to travel the mean free path distance λ between two collisions. Internally this distance λ is corrected for a small distance depending of the impact parameter of the collision. In the following we use λ for simplicity. The time Δt to travel the distance λ is given as

$$\Delta t = \frac{\lambda}{\langle \upsilon \rangle} \tag{1.44}$$

The average velocity $\langle \upsilon \rangle$ is calculated as mean value between the particle velocity of particle with mass m and Energy E_{part} at the beginning of path λ and the velocity of particle with mass m and Energy $E_{help} = E_{part} - E_{el}$ at the end of path λ. Her the electronic energy loss is subtracted. We find

$$<\upsilon>=\frac{1}{2}\sqrt{\frac{2}{m}}\left(\sqrt{E_{part}}-\sqrt{E_{help}}\right) \tag{1.45}$$

In subroutine inelast.F90 this formula is used to calculate the time step Δt. The cumulative time of an event after time zero of projectile impact is the calculated as

$$t=\sum_i \Delta t_i = \sum_i \frac{2\lambda_i}{\sqrt{\frac{2}{m}}} \frac{1}{\sqrt{E_{part,i}}+\sqrt{E_{help,i}}} \tag{1.46}$$

The pre-factor is defined in subroutine or_loss.F90, which is part of subroutine pots.F90. Her we define the speed of light in units [Å/fs] and the parameter mass_unit_cs = u·c^2 in units [eV].

$$\begin{aligned} c_{light} &= 2.99792458 \cdot 10^3 \quad Å / fs \\ uc^2 &= 9.315016246 \cdot 10^8 \quad eV \end{aligned} \tag{1.47}$$

Then we define the constant of a kinetic energy

$$eec = \frac{uc^2}{2c_{light}{}^2} = 51.82176674 \quad \left[\frac{eV \cdot fs^2}{Å^2} \right] \tag{1.48}$$

For a particle with mass m (in a.m.u) the rest mass is the given as

$$\frac{m}{2} = \frac{m_{amu} \cdot uc^2}{2c_{light}^2} = m_{amu} \cdot eec \quad \left[\frac{eV \cdot fs^2}{Å^2} \right] \tag{1.49}$$

Inserting into eq.(1.46) we get

$$t = \sum_i \Delta t_i = \sum_i \frac{2\lambda_i}{\sqrt{\frac{1}{m_{amu} eec}}} \frac{1}{\sqrt{E_{part,i}} + \sqrt{E_{help,i}}} \quad [fs] \tag{1.50}$$

This leads to

$$t = \sum_i \Delta t_i = \sqrt{4 m_{amu} eec} \sum_i \frac{\lambda_i}{\sqrt{E_{part,i}} + \sqrt{E_{help,i}}} \quad [fs] \tag{1.51}$$

The pre-factor is defined in subroutine or_loss.F90 and used in subroutine inelast.F90 as

$$tmpr(m_{amu}) = \sqrt{4 \cdot eec \cdot m_{amu}} \tag{1.52}$$

The time stamp output can be seen in the six output data files for backscattered projectiles, backsputtered recoils, transmitted projectiles, transmission sputtered recoils, stopped projectiles and stopped recoils. The time unit is [fs] and the time is given relative to the time zero of projectile impact. Typical elapsed time is several fs up to several ten fs.

4.20 Estimate of the average charge state of emitted projectiles

The equilibrium charge state of a fast projectile can be estimated by the formula

$$Z_{eff} = Z_1 \left(1 - \exp\left(-\frac{\upsilon}{\upsilon_0} Z^{-2/3} \right) \right) \tag{1.53}$$

The charge state varies between the atomic number of the projectile and zero, depending on the velocity of the projectile given by

$$\upsilon = \sqrt{\frac{2E_{kin}}{m}} \tag{1.54}$$

This velocity is compared to the Fermi velocity $\approx \frac{1}{200}c$ or the Bohr velocity $\approx \frac{1}{137}c$.

Examples are:

1 keV He: $\frac{\upsilon}{\upsilon_0} \approx 0.1$ and $Z_{eff} \approx 0.122$

1 MeV He: $\frac{\upsilon}{\upsilon_0} \approx 3.16$ and $Z_{eff} \approx 1.727$

10 MeV ^{11}B: $\frac{\upsilon}{\upsilon_0} \approx 6$ and $Z_{eff} \approx 4.36$

The time to change the charge state of an ion towards equilibrium can be determined from the experimental study “Charge Exchange Cross Sections for Helium Ions in Gases” by C. F. BARNETT AND P. M. STIER, Phys. Rev. 109 (1958) 385.

For 1 keV He ions with a velocity of 0.2187 nm/fs we find a cross section of transition from He$^+$ to He^0 in Ar gas of about $4 \cdot 10^{-16}$ cm^2/Ar gas atom. Adopting this, and using a typical atomic density of a solid of $5 \cdot 10^{22}$ atoms/cm^3 we get a characteristic length of 0.5 nm for neutralization. This corresponds to a characteristic time τ of 2.29 fs.

We can then use an estimate of the charge state as function of time

$$Z_{eff}(t) = Z_{start} \cdot \exp(-t/\tau) + \left(1-\exp(-t/\tau)\right) \cdot Z_{eff} \tag{1.55}$$

For the case of 1 keV He$^+$ ions we get $Z_{eff} \approx 0.1 - 0.7$ only for single and dual backscattering but only the equilibrium value $Z_{eff} < 0.1$ for multiple scattering

For the case of 500 keV He$^+$ ions we assume a characteristic length of 8 nm for double ionization. For a penetration path length of 4 nm (scattering near the surface) we find $Z_{eff} \approx 1.20$ with an equilibrium value $Z_{eff} \approx 1.5$. Therefore the 1+ charge state dominates over 2+ for backscattering near the surface.

4.21 Matrices of emitted particles as function of angles and energy

Compared to SDTrimSP the output of angular distributions of sputtered and scattered particles has be improved. The output can be defined more flexible by a number of input script file commands.

```
lmatrices = .true.      ! creates matrix output
matrix_e_min_r = 0.   ! min energy for recoiled (sputtered) particles
matrix_e_max_r = 500. ! max energy for recoiled (sputtered) particles
matrix_e_min_p = 0.   ! min energy for scattered projectiles
matrix_e_max_p = 5000.! max energy for scattered projectiles
matrix_e_steps_lin_r= 10 ! number of energy steps between e_min and e_max
                         ! for recoils
matrix_e_steps_lin_p= 10 ! number of energy steps between e_min and e_max
                         ! for projectiles
matrix_steps_pdecade_r= 4 !setting of steps for log energy scale, 4 per decade
matrix_steps_pdecade_p= 4 !setting of steps for log energy scale, 4 per decade
matrix_transmitted=.false.  ! create matrices for transmitted particles
matrix_backscattered=.true. ! create matrices for back-emitted particles
matrix_cumulative=.true.    ! cumulative or differential data collection
                              ! only for dynamic simulations
matrix_log_energy=.false.   !   logarithmic   energy   matrix   output(static)
matrix_cos_angle=.false.   ! cos instead of degrees for polar angles
```

For dynamic simulations four files for projectiles, recoils, back-emitted and transmitted particles are created. Each file has a 6digit extension, which is filled with and ascending number based on the number of histories in case of a dynamic simulation for each step of history output. Otherwise the 6 digit extension is zero.
Individual energy ranges and number of energy steps can be defined individually for recoils and projectiles. The log scales can be calculated in auto-mode for 9 linear steps per decade., starting at 0.1 eV.
The matrices consist of 30 columns of polar angles between 0 and 90 deg in steps of 3 deg and 60 rows of azimuthal angles between 0 and 180 deg in steps of 3 deg. In case of cos output instead of degrees of polar angles, 40 columns between 0 and 1 in steps of 0.025 are created.

The matrix output filenames are (r for recoils, p for projectiles):

```
  meagb_p_file ='matrix_energy_angle_back_proj.dat'
  meagt_p_file ='matrix_energy_angle_trans_proj.dat'
  meagb_r_file ='matrix_energy_angle_back_sputt.dat'
  meagt_r_file ='matrix_energy_angle_trans_sputt.dat'
  mepb_p_file ='matrix_energy_path_back_proj.dat'
  mept_p_file ='matrix_energy_path_trans_proj.dat'
  mpe_ex_p_file='matrix_start energy_vs max_penetration_back.dat'
  polar and azimuthal angular_matrix_file_br ='meagb_r_000000.dat'
  polar and azimuthal angular_matrix_file_bp ='meagb_p_000000.dat'
  polar and azimuthal angular_matrix_file_tr ='meagt_r_000000.dat'
  polar and azimuthal angular_matrix_file_tp ='meagt_p_000000.dat'
```

In addition to angular distributions of emitted particles, the program creates the files containing energy versus path relations.

In SDTrimSP the log scale conversion into increments istep has historical origin. There one uses the relation

$$istep = 12 \cdot \log_{10}(10 \cdot E) + 2, \tag{1.56}$$

With minimum energy E of 0.1 eV and the inverse:

$$E = 10^{\left(\frac{istep-2}{12}-1\right)} \tag{1.57}$$

The integer parameter istep ranges from 2 for E_{min}=0.1 eV up to 86 for E_{max}= 1MeV). In the matrix postprocessing routine matrix4.F90 one uses a parameter temp_nh which is a normalization factor for nh = number_histories * number_projectiles. There a factor

$$temp\_nh = \left(10^{\frac{1}{12}} - 1\right) 10^{-\frac{7}{6}} \cdot number\_histories \cdot number\_projectiles \tag{1.58}$$

appears. The factor temp_nh normalizes the emitted energy to a value per projectiles and the exotic prefactor is from an integral over 1 logarithmic energy increment. However, there seems a missing factor 12/ln(10) *10**(-7/6).

The new IMINTDYN program uses a more flexible way to define the linear and logarithmic energy steps for energies of backscattered, back sputtered and transmitted particles. First one defines the minimum and maximum energies, E_{min} and E_{max}, for the matrix output, individually for projectiles and recoils. The one defines the number of linear steps and the number of steps per decade n_{decade} for the log energy matrix output. Again, individually for projectiles and recoils. The minimum energy must not exceed 0.1 eV, so E_{min} is determined as max(0.1, E_{min}). The average energy <E> for a logarithmic energy step is calculated by integration, using the total number of logarithmic steps n_{total}

$$i = \mathrm{int}\left[\log_{10}\frac{E_{\max}}{E_{\min}}\right] \quad ; n_{total} = \mathrm{int}\left[\frac{E_{\max}}{10^i \cdot E_{\min}}\right] + n_{decade} \cdot i \quad , \tag{1.59}$$

the integer index n of a logarithmic step for given energy E

$$n = \frac{E}{10^i E_{\min}} + n_{decade} \cdot \mathrm{int}\left[\log_{10}\left(\frac{E}{E_{\min}}\right)\right] \quad , \tag{1.60}$$

The energy at the lower boundary of a logarithmic step i9nterval

$$E_i = 10^{\log_{10}(E_{\min})} + (i-1)\frac{1}{n_{total}}\log_{10}\left(\frac{E_{\max}}{E_{\min}}\right). \tag{1.61}$$

The average energy <E> is then obtained from Ei by integration for one interval {I,i+1}.

$$< E >= E_i \cdot fact_{log} \quad ; \quad with$$
$$fact_{\log} = \frac{10^D - 1}{D \ln 10} \quad and \quad D = \frac{1}{n_{total}}\log_{10}\left(\frac{E_{\max}}{E_{\min}}\right) \tag{1.62}$$

Postprocessing of the matrix output files is done using the program read_imindyn_matrix.F90. This program creates a new folder “postprocessing matrices” and fills this folder with a large number of individual matrix files. The input files are

```
meagb_p000000.dat
meagt_p000000.dat
meagb_r000000.dat
meagt_r000000.dat
```

with

..b_p..	for backscattered projectiles
..t_p..	for transmitted projectiles
..b_r..	for back-sputtered recoil
..t_r..	for transmitted-sputtered recoil

The program output is:

Internal index iout_pol = 0 : output of polar angles in degree
Internal index iout_e = 0 : output with linear energy scale of emitted particles

1.1 matrix__ag.. -> polar and azimuthal angle (number(integer))
1.2 matrix__ea.. -> energy and polar angle (number(integer))
1.3 matrix__eg.. -> energy and azimuthal angle (number(integer))
1.4 matrix__ee.. -> polar and azimuthal angle (energy(real))

The solid angle for an angular matrix element is defined as

$$\Delta\Omega = \int_{\varphi_1}^{\varphi_2}\int_{\vartheta_1}^{\vartheta_2} \sin\vartheta d\vartheta d\varphi = \Delta\varphi\left(\cos(\vartheta_2) - \cos(\vartheta_2)\right) \tag{1.63}$$

If the polar angle is expressed as its cosine value with constant increments of the cosine scale, then each matrix element $\Delta\cos(\vartheta)\Delta\varphi$ represents the same solid angle. In this case no further correction is necessary.
If the polar angle is expressed in degrees or radians, then we get

$$\delta\Omega = \sin\vartheta d\vartheta d\varphi = \sin\langle\vartheta\rangle(\vartheta_2 - \vartheta_1)(\varphi_2 - \varphi_2)\,. \tag{1.64}$$

The average polar angle $\langle\vartheta\rangle$ for a given matrix element is then

$$\langle\vartheta\rangle = \vartheta_2 - \frac{\vartheta_2 - \vartheta_1}{2}\,. \tag{1.65}$$

Each matrix element with angular range $\vartheta_2 - \vartheta_1$ and $\varphi_2 - \varphi_1$ covers a different solid angle. In order to normalize for identical solid angles, the content of a matrix element with upper boundaries (ϑ_2, φ_2) has to modified by

$$M(\vartheta_2,\varphi_2) \Rightarrow \frac{M(\vartheta_2,\varphi_2)}{\sin\left(\vartheta_2 - \frac{\vartheta_2 - \vartheta_1}{2}\right)} \tag{1.66}$$

The solid angles for each matrix element is small for small polar angles because the azimuthal angular steps become small.

In SDTrimSP the normalization is done by dividing by the total solid angle of each matrix element, and by dividing by the total number of projectiles (number of histories x number of projectiles per history) leading to a normalized matrix content per solid angle and per projectile. Such a normalization is not meaningful because it just equally reduces the values of the matrix content, but without getting a normalization such that the integrated matrix content is 2p, the solid angles of a hemisphere (The values still contain information about the differential sputter yields in case of recoils).

IMINTDYN uses the correction given by eq. (1.66). The correction according to eq. (1.66) increases the statistical error of a matrix element from $\sqrt{counts}$ to $\sqrt{counts/\sin\langle\vartheta\rangle}$. This can be corrected by averaging the counts for neighboring matrix elements M(i,j) (in azimuthal direction i) over ± m elements. Each element contributes with a fraction of 1/(2m+1) and

$$M_{smooth}(m_0,j)=\frac{1}{(2m+1)}\sum_{i=m_0-m}^{i=m_0+m}M(i,j). \tag{1.67}$$

The sum should run for $m=\max\left(1,\mathrm{int}\left(\frac{1}{2}\frac{1}{\sin\langle\vartheta\rangle}\right)\right)$.

M_{smooth} has the statistical error $\sqrt{counts/\sin\langle\vartheta_j\rangle}$. To maintain the statistical error $\sqrt{counts}$ after correction, we calculate the deviation from $M_{smooth}(m_0,j)$

$$M(m_0,j)-M_{smooth}(m_0,j)=M(m_0,j)-\frac{1}{2m+1}\sum_{i=m_0-m}^{i=m_0+m}M(i,j), \tag{1.68}$$

and use

$$\begin{aligned}M_{new}(m_0,j)&=M_{smooth}(m_0,j)+\frac{M(m_0,j)-M_{smooth}(m_0,j)}{\sqrt{1./\sin\langle\vartheta\rangle}}\\ M_{new}(m_0,j)&=M_{smooth}(m_0,j)+\left(M(m_0,j)-M_{smooth}(m_0,j)\right)\sqrt{\sin\langle\vartheta\rangle}\end{aligned}. \tag{1.69}$$

M(i,j) is an array with i_{max} and j_{max} elements. Averaging is done in a cyclic way, however, the azimuthal range is 0-180°. Therefore, we need a reflection of the index at the boundaries

$$\begin{aligned}&\{i-m,i+m\}\quad with\\ &i-m<i_{\min}=1:\quad i\to 1-(i+m);\\ &i+m>i_{max}:\quad i\to 2i_{\max}+1-(i+m);\end{aligned} \tag{1.70}$$

For typical 3° steps of polar angles, the averaging range is

0–3° : m=19 3-6°: m=6 6-9°: m=3 9-15°: m=2 15-30° : m=1

30-90°: m=0

This method does not work for the smallest polar angles 0-3°, 3-6° and 6-9°. The reason is that the counts per matrix element are usually small or even zero. Therefore, instead of dividing by the small value $\sin(\langle\vartheta\rangle)$ we add to the ±m neighboring matrix elements of element M(i,j) with the value of M(i,j) multiplied with a correction factor. This leads to a smoothing of the M(i,j) elements for azimuthal angles.

The correction factor is necessary due to the typical low count rate in matrix elements with very small solid angles (polar angles less than 9° or polar angles indices 1-3). Here the Poisson distribution applies and for low overall events, detected events are underestimated. The correction is necessary to reach the overall average counts compared to the polar angle regime 9-18° (polar indices j=4,5,6).

To account for this influence, we determine the relative solid angle for each matrix element with polar angle index j

$$\Omega_{rel}(j) = \frac{\sin\left(\langle\vartheta_j\rangle\right)}{\sin\left(\langle\vartheta_1\rangle\right)} \tag{1.71}$$

We then calculate the average raw count rate for polar angle indexes j=4,5,6 integrated for all azimuthal angles i=1,..,n_a with n_a=60.

$$\langle I_{4-6}\rangle = \frac{1}{3}\sum_{j=4}^{6}\frac{\sum_{i=1}^{na} M(i,j)}{\Omega_{rel}(j)} \tag{1.72}$$

The correction factor for polar angle index j is then given as

$$f_{corr}(j) = \frac{\sum_{i=1}^{na} M(i,j)}{\langle I_{4-6}\rangle} \tag{1.73}$$

Depending on the integrated counts for each polar index value j, the correction factor is 2.5-4 for polar angles 0-3°, 1.1-1.3 for polar angles 3-6°, and 1.0-1.15 for polar angles 6-9°.

IMINTDYN corrects the matrices with polar angles > 9° by correcting the statistical error according to eq.(1.69). Each matrix element then represents the same solid angle. And the content has the correct statistical error $\sqrt{counts}$.

The matrices E(i,j) with content "energy" are corrected in the same way.

$$E_{new}(m_0, j) = E_{smooth}(m_0, j) + \frac{E(m_0, j) - E_{smooth}(m_0, j)}{\sqrt{1./\sin\langle\vartheta\rangle}} \quad . \qquad (1.74)$$

$$E_{new}(m_0, j) = E_{smooth}(m_0, j) + \left(E(m_0, j) - E_{smooth}(m_0, j)\right)\sqrt{1./\sin\langle\vartheta\rangle}$$

The content of each matrix element then refers to an identical solid angle.

Internal index iout_pol= 0 ..degree (normalized with eq. (1.66))
Internal index iout_e = 0 ..lin
1.1 matrix__ag.. -> polar and azimut (counts) (normalized with eq. (1.66))
1.2 matrix__ea.. -> energy and polar (counts) (normalized with eq. (1.66))
1.3 matrix__eg.. -> energy and azimut counts)
1.4 matrix__ee.. -> polar and azimut (energy(real))

Internal index iout_pol = 0 : output of polar angles in degree
Internal index iout_e = 1 : output with logarithmic energy scale of emitted particles
2.2 matrix_lea.. -> log(energy) and polar angle (counts) (normalized with eq. (1.66))
2.3 matrix_leg.. -> log(energy) and azimuthal angle (counts) (normalized with eq. (1.66))

Internal index iout_pol =1 : output of polar angles as cos(angle)
Internal index iout_e = 0 : output with linear energy scale of emitted particles
3.1 matrixc_ag.. -> cos_polar and azimuthal angle (counts)
3.2 matrixc_ea.. -> energy and cos_polar (counts)
3.4 matrixc_ee.. -> cos_polar and azimuthal angle (energy(real))

Internal index iout_pol =1 : output of polar angles as cos(angle)
Internal index iout_e = 1 : output with logarithmic energy scale of emitted particles
4.2 matrixclea.. -> log(energy) and cos_polar (counts)

In simulation mode “static_simulation”, the option lmatrices=.true. also creates the following matrix files:
5.1.mepb_p_file ='matrix_energy_path_back_proj.dat'
energy versus path length for backscattered projectiles
5.2. mept_p_file ='matrix_energy_path_trans_proj.dat'
energy versus path length for transmitted projectiles
5.3 mpe_ex_p_file='matrix_energy_max_penetration_back.dat'
energy versus penetration depth for backscattered projectiles
5.4 morigin_ex_br ='matrix_energy_depthorig_back_sputt.dat'
energy versus depth of origin for backssputtered recoils
5.5 morigin_ex_tr ='matrix_energy_depthorig_trans_sputt.dat'
energy versus depth of origin for transmission sputtered recoil

4.22 Advanced book keeping options

IMINTDYN provides advanced book keeping options and better structured ASCII output files. Typically, each output file has a number of header rows with specific information about the simulation. Then up to 3 sub-header lines follow defining the title, unit and comment of the subsequent data columns. The sub-header lines are followed by numeric ASCII output. This file structure can easily read using the OriginLab ORIGIN® software for further data processing and graphics generation.

The imint_output_sputter and imint_output_target files are post processed using the program read_imintdyn_target.exe. The program extracts a large number of specific files on concentration distributions, sputtering yields etc. in a new created folder "postprocessing_data".

If calculation of angular and energy distributions is enabled using the command "lmatrices = .true", then a number of matrix files are created for projectiles and recoils as well as for transmitted and backscattered particles. These files with raw data a then post-processed by the program "read_imintdyn_matrix.exe" and a large number of specific output file distributions are stored in the new created directory "postprocessing_matrices". During postprocessing all matrix element data $M(\theta,\varphi)$ of angular data are normalized for identical solid angle and adjusted for a statistical error of $\sqrt{M(\theta,\varphi)}$.

IMINTDYN also creates detailed output files for each event of back-sputtered and forward sputtered recoils, back-reflected and forward transmitted projectiles as well as stopped recoils and projectiles. The corresponding files are

- partic_back_p_nnn.dat : backscattered projectile data
- partic_tran_p_nnn.dat : transmitted projectile data
- partic_back_r_nnn.dat : backsputtered recoil data
- partic_tran_r_nnn.dat : transmission sputtered recoil data
- partic_stop_p_nnn.dat : stopped projectile data
- partic_stop_r_nnn.dat : stopped recoil data

Compared to SDTrimSP these files have a 3 digit index nnn, which is used for numbering the data files in case on a series simulation (series of energies or angles). The files also contain several additional columns.

Columns 1 and 2 show the processor id and the projectile counter for each projectiles. With these numbers a recoil can be associated with a specific projectile, which is necessary for coincidence analysis of scattering/recoil events.

Column 3 of recoil data shows recoil atomic index.

Column4 of recoil data shows the generation of the collision cascade. Generation 1 is the projectile, generation 2 the first recoil event, etc.

Column4 of projectiles data shows the atomic index of the collision partner of the collision cascade. Generation 1 is the projectile, generation 2 the first recoil event, etc.

Colum 5-7 list the total number of collisions of the respective particle as well as collisions with local scattering angle larger than a medium angle and larger than a huge angle. These data can be used as a filter to select specific recoils.

The last three columns are a weight factor, and the cosine values of the largest medium scattering angle and the largest huge scattering angle. The weight factor corrects scattering cross sections in case of enforced scattering and no-Rutherford cross sections. A scattering spectrum (i.e. counts versus energy) extracted from the data file can then be multiplied with the weight factor to obtain a spectrum reflecting correct scattering cross sections. This correction can be adjusted by calculating an average count rate and correcting the statistical error using the square root of the weight factor.

4.23 Post processing programs for creating ion scattering spectra

The data output files described in the previous chapter can be processed to calculate ion scattering spectra for CERDA, EBS, ERDA and ERCS scattering geometries as well as NRA.

For CERDA and ERCS geometry the coincidence spectra are calculated form the data files

“partic_trans_p_nnn.dat” and “partic_trans_r_nnn.dat”

For EBS geometry, the data file “partic_back_p_nnn.dat” is analyzed

For ERDA geometry, the data file “partic_back_r_nnn.dat is analyzed.

For NRA geometry, the data file “partic_back_r_nnn.dat is analyzed.

The postprocessing program allows to define a bin size for the spectra (default 1024 bins for the energy of scattered ions) and a limitation for the scattering angle regime.

The program the calculated energy histograms and corrects for cross section weight factors for different elements, different scattering energies and impact parameters. Three different histograms are calculated with extensions

- 1h0m : 1 huge scattering angle and no medium scattering angle, a spectrum which contains single scattering events
- 1h1m : 1 huge scattering angle and one medium scattering angle, a spectrum which describes dual scattering events
- xhxm : > 1 huge scattering angle and > 1 medium scattering angle, a spectrum which contains multiple scattering events

In case of a NRA simulation, there is no huge collision but instead the nuclear reaction. The hree different histograms are calculated with extensions are

- 0h0m : 1 nuclear reaction and no medium scattering angle, a spectrum which contains single nuclear reaction events

- 0h1m : 1 nuclear reaction and one medium scattering angle, a spectrum which resembles dual scattering
- 0hxm : >1 nuclear reaction and >1 medium scattering angle, a spectrum which contains multiple scattering events

The selection of huge angle and medium angles is done in the IMINTDYN input script file. The number of corresponding scattering events are tabulated in the data output files.

In addition to the raw histograms, which do not take the cross-section weighting factors into account, the program also calculated the weighted spectra. In this case the raw spectra $N_{raw}(E)$ are multiplied with individual weighting factors $w(E)$ for each scattering event. This would lead to fluctuations around the average value in the resulting histogram with count rate $\left\langle w(E) \cdot N(E) \right\rangle \pm w(E)\sqrt{N(E)}$ exceeding the expected statistical error $\sqrt{w(E) \cdot N(E)}$. To correct the spectra, we apply a smoothing over a few bins (typically ±2 bins) to obtain $\left\langle w(E) \cdot N(E) \right\rangle$ and calculate the weighted count rate as

$$N_w(E) = \left\langle w(E) \cdot N(E) \right\rangle + \frac{w(E) \cdot N(E) - \left\langle w(E) \cdot N(E) \right\rangle}{\sqrt{w(E)}} . \tag{1.75}$$

The operation leads to the correct weighted average value $\left\langle w(E) \cdot N(E) \right\rangle$ and the expected statistical error $\sqrt{w(E) \cdot N(E)}$.

4.24 ESA detector simulation

An ESA detector has an energy resolution given as

$$\frac{\Delta E}{E} = \text{constant}, \text{ typically } 0.5\% \tag{1.76}$$

The spectrum is sorted into energy bins of equal bin width ΔB, which is typically smaller than the energy resolution ΔE for sufficient high particle energies E. Depending on the detected energy E, a scattering event has to be distributed into one out of several energy bins around the central bin which corresponds to E. This is done in the postprocessing mode -ESA for backscattering spectra using random numbers.

For the backscattering postprocessing program we have implemented the command line option

```
-esamode resolution
- e resolution
```

With resolution given as dimensionless ratio ΔE/E.

In this mode, the energy resolution is calculated and the event is distribution by random number to an energy bin accessible by the calculated energy resolution.

4.25 TOF detector simulation

A TOF detector is defined by its flight path l_{TOF} and the time resolution Δt. A particle with kinetic energy $E = \frac{1}{2} M \upsilon^2$ passes the flight path within a time

$$t(E) = \frac{l_{TOF}}{\upsilon} = \frac{l_{TOF}}{\sqrt{2E/M}} . \tag{1.77}$$

The flight time through the analyzer is a function of energy and mass.

From this we get for a fixed mass particle

$$t(E) \pm \Delta t/2 = \frac{l_{TOF}}{\sqrt{2(E \mp \Delta E/2)/M}} . \tag{1.78}$$

`The energy resolution of a TOF system is then

$$\begin{aligned} 2(E \mp \Delta E/2)/M &= \left(\frac{l_{TOF}}{t(E) \pm \Delta t/2} \right)^2 \\ (E \mp \Delta E/2) &= \frac{M}{2} \left(\frac{l_{TOF}}{t(E) \pm \Delta t/2} \right)^2 \\ \Delta E &= \frac{M}{2} \left(\frac{l_{TOF}}{t(E) - \Delta t/2} \right)^2 - \frac{M}{2} \left(\frac{l_{TOF}}{t(E) + \Delta t/2} \right)^2 \end{aligned} \tag{1.79}$$

Here we subtracted the terms with different sign.

$$\Delta E=\frac{M}{2}l_{TOF}^2\frac{2E}{M}\left[\left(\frac{1}{l_{TOF}-\sqrt{2E/M}\Delta t/2}\right)^2-\left(\frac{1}{l_{TOF}+\sqrt{2E/M}\Delta t/2}\right)^2\right]$$

$$\Delta E=El_{TOF}^2\left[\frac{1}{\left(l_{TOF}-\sqrt{2E/M}\Delta t/2\right)^2}-\frac{1}{\left(l_{TOF}+\sqrt{2E/M}\Delta t/2\right)^2}\right]$$

$$\Delta E=El_{TOF}^2\left[\frac{\left(l_{TOF}+\sqrt{2E/M}\Delta t/2\right)^2-\left(l_{TOF}-\sqrt{2E/M}\Delta t/2\right)^2}{\left(l_{TOF}-\sqrt{2E/M}\Delta t/2\right)^2\left(l_{TOF}+\sqrt{2E/M}\Delta t/2\right)^2}\right]$$

$$\Delta E=El_{TOF}^2\left[\frac{l_{TOF}\sqrt{2E/M}\Delta t}{\left(l_{TOF}^2-\left(E(\Delta t)^2/2M\right)\right)^2}\right]=\frac{\sqrt{2}E^{3/2}l_{TOF}^3\Delta t}{\sqrt{M}\,l_{TOF}^4}\left[\frac{1}{1-\left(E(\Delta t)^2/2Ml_{TOF}^2\right)}\right]^2$$

(1.80)

$$\Delta E=\frac{\sqrt{2}E^{3/2}\Delta t}{\sqrt{M}\,l_{TOF}}\left[\frac{1}{1-\left(\frac{E}{2M}\frac{(\Delta t)^2}{l_{TOF}^2}\right)}\right]^2 \tag{1.81}$$

The characteristic parameter for a TOF system is the dimensionless ratio of time resolution and path length.

$$r=c\left(\frac{\Delta t}{l_{TOF}}\right) \tag{1.82}$$

$$\Delta E=rE\sqrt{\frac{2E}{Mc^2}}\left[\frac{1}{\left(1-\frac{Er^2}{2Mc^2}\right)^2}\right] \tag{1.83}$$

With Mc^2 the rest energy of the particle. For $\Delta t=300ps$ and $l_{TOF}=1m$ we get $r=8.99377\cdot10^{-2}$. The term in the square bracket is usually close to unity because Mc^2 is of the order of several GeV, E of the order of keV and r^2 << 1. Therefore we get

$$\Delta E\approx rE\sqrt{\frac{2E}{Mc^2}}\ . \tag{1.84}$$

Example: Oxygen with E=100 keV, $Mc^2 \approx 16 \cdot 10^9$ eV : $\Delta E \approx 32\, eV$

Or:

$$\frac{\Delta E}{E^{3/2}} \approx \sqrt{\frac{2r^2}{Mc^2}} \,. \tag{1.85}$$

The spectrum is sorted into energy bins of equal bin width ΔB, which is typically smaller than the energy resolution ΔE for sufficiently high particle energies E. Depending on the detected energy E, a scattering event has to be distributed into one out of several energy bins around the central bin which corresponds to E. This is done in the postprocessing mode -t or -tof for elastic recoil detection spectra using random numbers. The command line argument is followed by δt (time resolution) and l_{tof} (flight path). Default values are δt=300 ps and l_{tof}= 1m.

For backscattering and ERDA postprocessing programs we have implemented the command line options

```
-tofmode  resolution lenth
- t  resolution length
```

with resolution given in units {ps] and length of the flight path given in units [m].

In this mode, the energy resolution is calculated and the event is distributed by random number to an energy bin accessible by the calculated energy resolution.

5. References

[1] Low energy ion-solid interactions: a quantitative experimental verification of binary collision approximation simulations, Hans Hofsäss, Felix Junge, Patrick Kirscht and Koen Karl Frans van Stiphout, Material Research Express (2023) DOI 10.1088/2053-1591/ace41c

[2] Low energy ion-solid interactions: a quantitative experimental verification of binary collision approximation simulations, Hans Hofsäss, Felix Junge, Patrick Kirscht and

Koen Karl Frans van Stiphout, Material Research Express (2023) DOI 10.1088/2053-1591/ace41c

[3] Binary collision approximation simulations of ion solid interaction without the concept of surface binding energies, H. Hofsäss and A. Stegmaier, Nucl. Instr. Meth B 517 (2022) 49-62

[4] Mutzke, A., Schneider, R., Eckstein, W., Dohmen, R., Schmid, K., Toussaint, U. v., et al.(2019). *SDTrimSP Version 6.00* (IPP 2019-02). Garching: Max-Planck-Institut für Plasmaphysik. doi:10.17617/2.3026474.

[5] W. Eckstein, Computer Simulation of Ion-Solid Interactions (Springer, Berlin, 1991)

[6] https://www.intel.com/content/www/us/en/developer/tools/oneapi/overview.html

[7] MPI: A Message-Passing Interface Standard, Version 4.1, Message Passing Interface Forum, November 2, 2023, https://www.mpi-forum.org/docs/mpi-4.1/mpi41-report.pdf

[8] https://www.amd.com/de/products/cpu/amd-ryzen-threadripper-pro-7965wx

[9] J.P. Biersack and W. Eckstein, Appl. Phys. A34 (1984) 73.

[10] Prediction of ion-induced nanopattern formation using Monte Carlo simulations and comparison to experiments, H. Hofsäss and O. Bobes, Appl. Phys. Reviews 6 (2019) 021307

[11] W. Moeller and W. Eckstein, Nucl. Instr. Meth. B2 (1984) 814

[12] W. Moeller, W. Eckstein, J. P. Biersack, Computer Physics Comm. 51 (1988) 355.

[13] J. F. Ziegler, M. D. Ziegler, J. P. Biersack, Nucl. Instr. Meth. B268 (2010) 1818.

[14] W. Moeller, M. Posselt, TRIDYN_FZR User Manual, Wissenschaftlich Technische Berichte FZR-317, ISSN-1437-322X, (2001)

[15] A Cezairliyan, C. Y. Ho, A. C. Anderson, *Specific heat of solids*. (Hemisphere Publishing Corp., 1988).

[16] https://www-nds.iaea.org/exfor/ibandl.htm

[17] https://webbook.nist.gov/chemistry/ ; National Institute of Standard and Technology Chemistry Web Book, DOI: https://doi.org/10.18434/T4D303, latest update: 2018.

[18] E. Oyarzabal, R. P. Doerner, M. Shimada, and G. R. Tynan, J. appl. Phys. 104 (2008) 043305

6.