

# Graphical user interface for SDTrimSP to simulate sputtering, ion implantation and the dynamic effects of ion irradiation

P.S. Szabo<sup>a,b,\*</sup>, D. Weichselbaum<sup>a</sup>, H. Biber<sup>a</sup>, C. Cupak<sup>a</sup>, A. Mutzke<sup>c</sup>, R.A. Wilhelm<sup>a</sup>, F. Aumayr<sup>a</sup>

<sup>a</sup> Institute of Applied Physics, TU Wien, Vienna, Austria

<sup>b</sup> Space Sciences Laboratory, University of California, Berkeley, United States

<sup>c</sup> Max-Planck-Institute for Plasma Physics, Greifswald, Germany

## ARTICLE INFO

### Keywords:

SDTrimSP  
Graphical user interface  
Ion–solid interaction  
Sputtering  
Ion ranges  
Ion-induced damages  
simulation

## ABSTRACT

SDTrimSP is a popular simulation program to compute several effects of the interaction between an impinging ion and a solid, such as ion implantation ranges, damage formation or sputtering of surface atoms. We now introduce a graphical user interface for SDTrimSP to make its operation more accessible for a broad group of users. It is written as a separate Python program and is not restricted to any specific operating system. The interface allows a quick and easy start as well as the direct evaluation of SDTrimSP simulations. Its capabilities are demonstrated here in the form of several example cases, including the dynamic simulations with SDTrimSP, where ion-induced target changes are taken into account. The presented graphical user interface is made freely available to support a large number of users in performing simulations of ion–solid interaction.

## 1. Introduction

Quantitative understanding of the interaction between ions and solids is of continuous interest in several scientific fields. For semiconductor applications, ion beam techniques have become essential tools for precise material modification [1,2]. Due to the increasing demands on realizing small scale devices, the exact knowledge of ion ranges, energy depositions and sputtering effects during the interaction with the irradiated materials is extremely important. A valid description of these effects has also been of great interest for the understanding of radiation damage in nuclear materials [3–5]. For low energy ions, nuclear fusion research has been a strong driver for promoting experimental and numerical advances in understanding the erosion by ion bombardment. Here challenges are posed by the large fluences of plasma ions that hit the reactor wall material [6,7], as well as complex surfaces in both composition and roughness [8,9]. In planetary science, ions from the solar wind or from planetary atmospheres have significantly modified the surfaces of bodies such as the Moon or Mercury over billions of years [10,11]. Ion bombardment and other space weathering effects determine how we observe planetary surfaces today and their understanding will advance our knowledge of their history. Traditionally, ion irradiation of solids also has several analytical and applied purposes: Ions at different energy regimes are used for material characterization from Secondary Ion Mass Spectrometry (SIMS) to various Ion Beam Analysis (IBA) methods. Slow ion beams are a common tool for surface preparation or nanostructuring [12,13]. Furthermore, layer growth by

different sputtering deposition techniques have become essential for creating complex thin films for a broad field of applications [14–17].

For all these topics, numerical simulations are essential tools to describe how an ion interacts with a solid target upon impinging its surface. SDTrimSP represents a code that is well-suited for calculating the different effects that occur for all the above-described scenarios [18]. It is based on the Binary Collision Approximation (BCA), which simulates the large number of collisions that take place following the energy loss of the ion by only calculating a sequence of independent two-body collisions [19]. This method represents an alternative to, for example, Molecular Dynamics (MD) simulations. However, it is still able to produce accurate results at a fraction of the computational effort [20]. The dynamic capabilities of SDTrimSP further allow simulations that take into account erosion of different surface layers, surface concentration changes due to preferential sputtering and the development of depth profiles of implanted ions [18]. In such dynamic simulations, even different forms of diffusion effects can be included to describe the development of the sample composition more realistically [21,22].

SDTrimSP has regularly been found to accurately describe experimental findings and especially for the calculation of sputtering yields, its performance fares better in comparison to similar programs such as the SRIM software [23–26]. Compared to SRIM, SDTrimSP also has an advantage with its dynamic capabilities and it gives the user more control over most parameters of the simulation. However, in many cases,

\* Corresponding author at: Institute of Applied Physics, TU Wien, Vienna, Austria.  
E-mail address: [szabo@berkeley.edu](mailto:szabo@berkeley.edu) (P.S. Szabo).

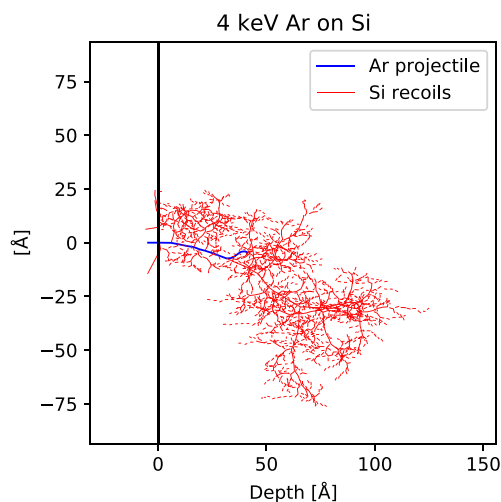


Fig. 1. As an example for a BCA simulation, the simulated particle trajectories of a single 4 keV Ar ion hitting a Si target under normal incidence are shown in an  $xz$ -projection. Results are from an SDTrimSP simulation, the projectile trajectory is plotted in blue and the recoil trajectories in red.

SRIM is still the standard choice for simulating ion–solid interaction. The main reason for this is its graphical user interface (GUI), which makes it easy to set up and visualize simulation results. Therefore, we now also present a GUI for SDTrimSP to increase its accessibility and make it easier to use for simulations of ion–solid interaction. As it is similarly implemented for the Elastic Recoil Detection Analysis software Potku [27], the GUI is realized in the form of an accompanying Python program that allows fast and easy pre- and post-processing of SDTrimSP simulations. It represents an additional tool compared to the default text-based inputs and outputs of the program. The presented GUI is made freely available with the aim of making it easier for a large number of users, including those from other research communities, to profit from SDTrimSP’s additional simulation capabilities.

## 2. Overview of the GUI

### 2.1. Background: BCA simulations of ion–solid interactions

BCA programs simulate the effects of ion irradiation of a solid target in the form of a large number of binary collisions. Collisions between the ion and target atoms as well as between different target atoms are sequentially calculated, Fig. 1 gives an example of the resulting particle trajectories in the solid. Most programs implement amorphous targets by moving a particle by a free path length and then choosing a random impact parameter for the scattering process. Scattering angles and energies can be calculated from an interaction potential, where typically screened Coulomb potentials such as the Krypton–Carbon (KrC) potential or the Ziegler–Biersack–Littmark (ZBL) potential are applied [28,29]. Inelastic energy loss by electronic stopping is implemented as a friction force that causes energy loss of particles between two collisions. Commonly implemented models for this energy loss include, for example, the Lindhard–Scharff model or the Oen–Robinson model at low energies [30,31]. In a stepwise manner, the projectile and all moving recoil are followed, until they leave the target again or their energy falls below a pre-defined threshold and they are regarded as stopped in the target. This procedure allows the simulation of ion implantation ranges, ion-induced damage distributions, reflection coefficients or sputtering yields. The quantitative capabilities of the simulations are strongly connected to relevant material properties that have to be taken from tabulated values. This includes especially the surface binding energy and displacement energy, which respectively influence sputtering and vacancy formation.

BCA codes have been applied for the simulation of the interaction between energetic ions and solids for several decades. The first approaches focused on computing ion ranges, ion-induced damages or channeling effects from simulating nuclear collisions [32–34]. In the MARLOWE code, binary collisions based on the Molière interatomic potential were combined with a modified Firsov model for the inelastic energy loss [19,35]. While MARLOWE takes into account a crystal structure, amorphous targets were implemented in the Transport of Ions in Matter (TRIM) program [36]. Combining low-energy electronic stopping models with Bethe–Bloch stopping, as well as adapting the free flight paths between collisions, improved the validity for simulating ion ranges in high-energy irradiations. TRIM was later expanded to a sputtering version TRIM.SP [37], which also includes a full treatment of recoil atoms that are set in motion by previous collisions. TRIM.SP allowed the simulation of sputtering yields as well as energy and angular distributions of sputtered particles in agreement with experimental results. A TRIM version is included in the SRIM (Stopping and Range of Ions in Matter) package [23], which still represents one of the most commonly used BCA programs. However, based on TRIM.SP further improvements were introduced in the dynamic version TRIDYN [38,39], making it possible to simulate ion-induced changes of the irradiated target such as preferential sputtering, projectile implantation or mixing of different layers. SDTrimSP was then derived from TRIDYN and includes capabilities of parallel computation [40]. Additional features from diffusion to chemical sputtering have since been added over time [18,21,22,41]. Furthermore, both TRIDYN and SDTrimSP have been expanded to separate programs TRI3DYN as well as SDTrimSP-2D and SDTrimSP-3D, which include 2D or 3D treatments of the surface morphology [42–44]. They enable the implementation of rough surfaces or nanostructures and simulate their effect on the ion irradiation as well as morphology changes as a result of the ion bombardment.

### 2.2. Background: SDTrimSP

The GUI presented in this article is developed for the use of the 1D version of SDTrimSP. As described above, it combines the capabilities of the classic TRIM versions with the possibilities of dynamic simulations, improved computational efficiency by parallelization and further new developments. It especially gives the user more control over several key parameters of the simulation. Simulations can be performed for a specific ion energy and incidence angle, but energy and angle distributions or mixed ion beams can be simulated as well. For the calculations, several integration methods and interaction potentials can be chosen. Especially the latter parameter is important, as the default KrC interaction potential in SDTrimSP was found to lead to better agreement with experiments than SRIM’s default ZBL potential [24]. Inelastic energy loss can be implemented from the Lindhard–Scharff model, the Oen–Robinson model, the Ziegler–Biersack model or several tabulated values.

SDTrimSP is written as a Fortran code that takes inputs from a “tri.inp” and other “.inp” text files, depending on the used case. The “tri.inp” file is formatted as a Fortran namelist that allows the direct assignment of variable values and includes the parameters for every ion–solid interaction simulation with SDTrimSP. More input files are used, for example, to define a sample with different layer compositions or energy and angle distributions of incoming ions. An SDTrimSP simulation can be started by running the compiled Fortran code in the folder where the “tri.inp” input file is stored. During a simulation run and after its completion, outputs are saved in several “.dat” text files.

### 2.3. Basic working principle of the GUI

The main goal of our GUI is to provide a more convenient method for creating inputs and evaluating results of SDTrimSP simulations, especially for inexperienced users. This is achieved with an easily

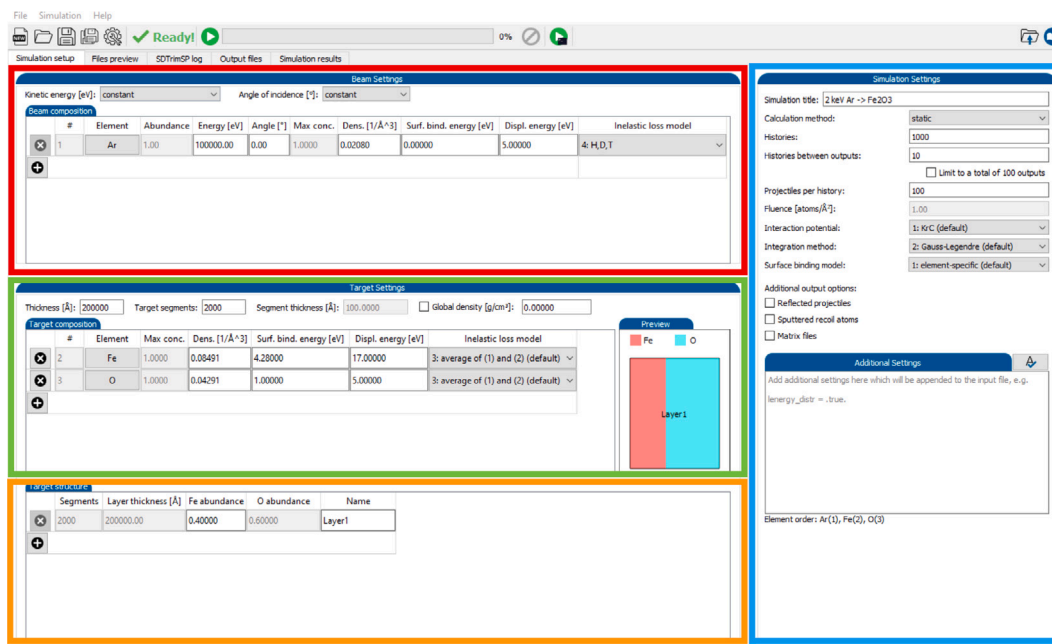


Fig. 2. The main view of the GUI for setting up important simulation parameters.

accessible interface for defining the key calculation parameters. Based on the chosen options, the GUI automatically creates the appropriate “tri.inp” file, as well as a “layer.inp” file if multiple target layers with different compositions are chosen. The simulation can be directly started from the GUI and its progress can be monitored in the GUI window as well. The interface also includes capabilities for the direct evaluation of several important SDTrimSP results. For static calculations, sputtering yields are compiled and ion ranges as well as displacements of target atoms can be plotted over depth in the sample. Dynamic simulations can be monitored while they are running to allow a quick visualization of the fluence dependence of sputtering yields, surface concentration changes or the composition over depth, to give examples of the available parameters. Furthermore, energy and angular distributions of reflected ions as well as sputtered atoms can be visualized.

The pre- and post-processing capabilities of the GUI are independent from each other and can thus also be used separately. In principle, the GUI can be used for only preparing input files that are executed on another machine and in the same manner, the GUI can also be applied for solely evaluating results of previously performed SDTrimSP simulations.

#### 2.4. The graphical user interface

The GUI companion program for SDTrimSP is written in Python 3 based on the PyQt5 package that implements the software toolkit Qt for creating graphical user interfaces with Python. This manuscript describes the version 1.0 of the GUI, which is the up-to-date version as of March 2022 with further improvements being planned for the future. The GUI has been tested under Python 3.7, 3.8 and 3.9 on Linux, Windows and Mac operating systems. It supports SDTrimSP versions 6.01 and newer, with 6.06 being the latest available version at the time of the publication of this article.

The GUI is mainly organized in five tabs that separate different functionalities, which are described in the following paragraphs.

The **Simulation setup** view (see Fig. 2) is the screen that is visible upon startup of the program and gives the user the possibility of defining the parameters of the SDTrimSP simulation. Under “Beam Settings” (red box in Fig. 2) ion beam species, energies and incidence angles can be defined. Elements can be chosen from a periodic table as

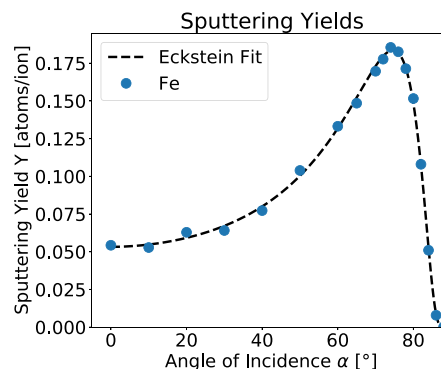


Fig. 3. Angular dependence of the Fe sputtering yield under 500 eV D irradiation.

it is similarly realized in the SRIM interface [23]. For each element, parameters of the incident ions can be defined, most importantly impact energy and angle as well as the concentration in the ion beam. Furthermore, the maximum target concentration for implanted ions in a dynamic simulation can be set and one of the six possible options for inelastic energy loss can be chosen. For the latter parameter, as well as for many other options in general, tooltips can be accessed by hovering the mouse over the specific entries to receive more information on the respective options.

Under “Target settings” (green box) the target elements can be defined similarly with the additional option of defining custom atomic densities. As a new feature specific to the GUI, setting one global density is possible to better implement realistic properties for composite materials, similar to how it is realized in SRIM [23]. Using this possibility should be kept in mind because the density input can significantly affect some simulation outcomes. We illustrate this here with the example of  $\text{Mg}_2\text{SiO}_4$  (olivine), which is commonly found on planetary bodies such as the Moon [45], where it is exposed to continuous plasma precipitation. The default atomic density in SDTrimSP for its composition would be  $0.044 \text{ atoms}/\text{\AA}^3$  and changes substantially to  $0.098 \text{ atoms}/\text{\AA}^3$ , when a realistic material density of  $3.27 \text{ g}/\text{cm}^3$  is used with the GUI. For solar wind H ions at 1 keV this affects sputtering

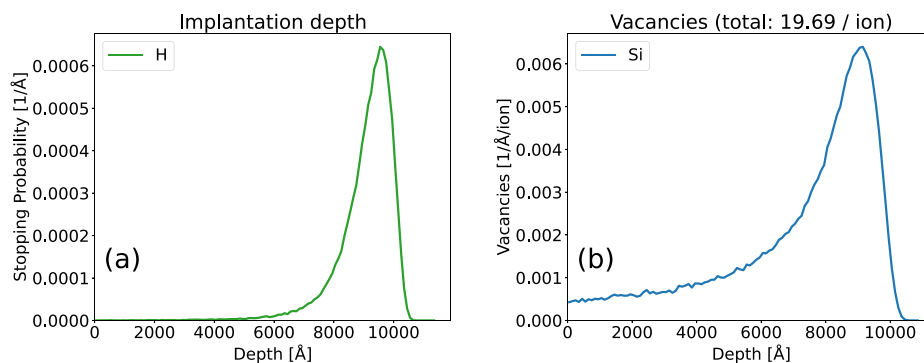


Fig. 4. Implantation depth (a) and ion-induced vacancies (b) for a 100 keV H irradiation of a Si target under normal incidence.

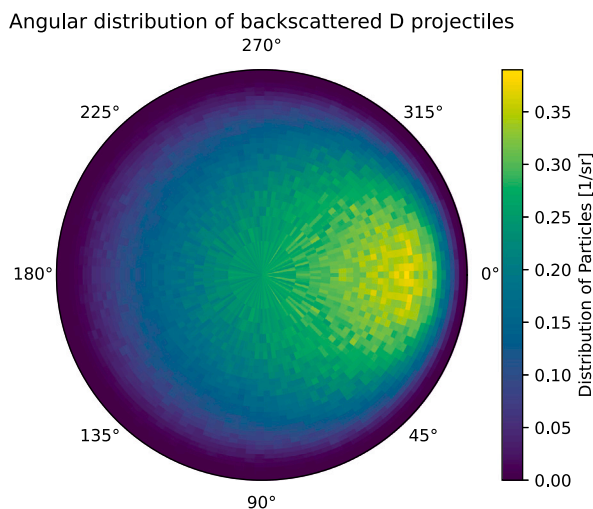


Fig. 5. Angular distribution of reflected D ions for a 500 eV D irradiation of an Fe target under an incidence angle of  $60^\circ$ . In this plot, ions hit the target from the  $180^\circ$  direction. The data is binned, so that each bin has the same solid angle size.

yields and implantation depths by up to 60% with otherwise identical simulation parameters.

The exact composition of the target can be defined in “Target structure” (orange box), where multiple layers with different thickness and compositions can be added. This enables, for example, multilayer thin-film samples or oxide layers to be included in the simulation. A preview of the defined structure is visible next to the target-related tables under “Preview”. “Simulation Settings” on the right side of the window (blue box) give control over simulation parameters such as choosing static or dynamic modes, the statistic parameters that determine the accuracy of the simulation or the fluence of a dynamic simulation. Interaction potentials, integration methods and surface binding energy models can be chosen as well from the options included in SDTrimSP. Additional output possibilities are given: For example, toggling “Reflected projectiles” and “Sputtered recoil atoms” activates separate outputs of emission data for each ion that is reflected or each atom that is sputtered in the simulation respectively. This is done as a list with one entry for every single ion and due to file size considerations, the activation of these options is only recommended if data stored for each single particle are of interest.

The given options include the most important parameters that have to be defined for every simulation or allow options that are of interest for the majority of users. Detailed control over several more advanced capabilities of SDTrimSP that might only be relevant for specific use cases, from different diffusion effects to chemical sputtering of C by H ions, has not yet been included. The GUI program is made open source

and can therefore be extended to other users specific requirements. For now, we have refrained from adding direct access to too many specific options, in an effort to not overload the main GUI window. However, the text box “Additional Settings” allows manually adding any further desired parameters as they would be written in the “tri.inp” file. Some of those options are dependent on the order of the used elements, which is given below the input box. For more information on such additional SDTrimSP options, the SDTrimSP documentation (see [18]) can directly be accessed over the book icon in the upper right corner of the GUI.

When the next tab **Files preview** is selected, the input files “tri.inp” and “layer.inp” are shown as they are created from the chosen inputs. After the setup of the simulation, SDTrimSP can directly be executed with the green “Play” button that starts SDTrimSP via a process that is connected to the GUI program. This allows direct feedback on the current simulation status, but SDTrimSP will be aborted if the GUI window is closed. If this is not desired, an additional detached mode is available (green play button with a small console logo), which runs SDTrimSP in a separate console window. If SDTrimSP is not started in the detached mode, the direct console output can be accessed in the tab **SDTrimSP log**. The next tab **Output files** gives a list of all created output “.dat” files and allows previewing or opening these files in a separate text editor.

The main capability for post-processing of SDTrimSP simulations is found under **Simulation results**, which is separated into a list of result values and plot options as well as a plotting window. By selecting a list entry on the left, the respective plot is shown on the right. For polar plots of particle distributions and fluence-dependent results of dynamic simulations, data can already be plotted while the simulation is still running. The relevant output files are re-read by the GUI program continuously and the plots are automatically refreshed to include the latest outputs. The plot images can directly be saved from the GUI and this can also be done for the outputs that are plotted: By pressing the “save” button next to the “Available data for plotting” header, a text file with all shown data can be exported. These text files can then be included in other post-processing software to compare, for example, different simulations or numerical and experimental data.

Both the plot images themselves as well as the data that is used to create them can directly be saved for later use from the GUI. As mentioned before, graphical evaluation of SDTrimSP results works for any simulation and not just those that have been set up and started with the GUI itself.

## 2.5. Availability

The GUI is publicly available online on github (<https://github.com/psszabo/SDTrimSP-GUI>) and distributed under the GPLv3 license. One condition for usage of the GUI program is that this manuscript is cited in any publication that presents results achieved with the GUI.



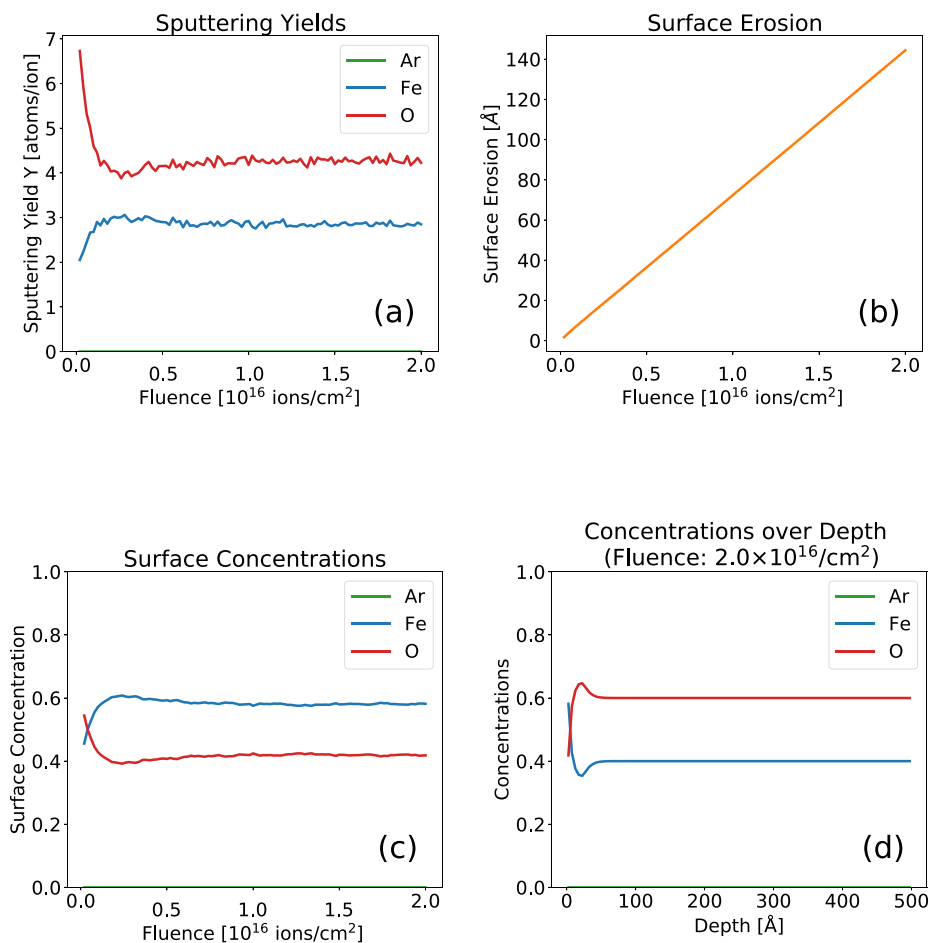


Fig. 6. Fluence-dependent sputtering yields (a), surface erosion (b) and surface concentrations (c), as well as depth-dependent concentrations at the end of the irradiation (d) of a  $\text{Fe}_2\text{O}_3$  target with 2 keV Ar ions under  $60^\circ$ .

### 3. Examples for usage cases

The evaluation capabilities of the GUI mainly focus around the direct plotting of different results to allow a well-arranged overview of the simulation outcomes. Depending on the chosen calculation mode and the toggled output options, different plots become available, with some examples being given here. The plots are directly saved from the GUI, without any further processing, apart from increasing font sizes for improving the readability in the manuscript. All shown examples were simulated with SDTrimSP 6.01 and the used input files are available from the supplementary data of this manuscript or at <https://github.com/psszabo/SDTrimSP-GUI>.

#### 3.1. Static simulations

Fig. 3 shows the angular dependence of the sputtering yield for a 500 eV D irradiation of an Fe target from a static calculation, which means that changes to the target are not included in the simulation. For the calculation shown in Fig. 3, the “full angular dependence” option for the angle of incidence was chosen, upon which SDTrimSP automatically runs multiple simulations at different incidence angles. A fit with the formula by Eckstein and Preuss is automatically included [46], represented by the black dashed line. The derived fit parameters are shown in the GUI and stored with the simulation results, when the plot data is saved.

For the normal incidence simulation of 100 keV H ions hitting a Si target, the plots in Fig. 4 depict the depth of implantation and the ion-induced vacancies as calculated by SDTrimSP. It can be nicely seen that

the peaks for both distributions are at a depth close to  $1 \mu\text{m}$ . However, some vacancies are also created closer to the surface due to collisions that transfer enough energy to overcome the displacement energy of the target atoms.

By enabling the options for additional information on reflected ions and sputtered atoms, their energy and angular distributions can also be analyzed. Fig. 5 gives the example of reflected D ions from an Fe target, with an incident energy of 500 eV and an incident angle of  $60^\circ$  (in the plot, the ions come from the left at an azimuthal angle of  $180^\circ$ ). The forward scattering dominance at azimuth angles of around  $0^\circ$  is evident, but due to the multiple scattering events inside the solid a very diffuse distribution is observed.

#### 3.2. Dynamic simulations

The dynamic simulations of SDTrimSP allow the computation of changes of the target due to the ion bombardment. Fig. 6 shows how these effects can be visualized with the GUI for the case of 2 keV Ar ions sputtering a  $\text{Fe}_2\text{O}_3$  target under  $60^\circ$ , with a simulated fluence of  $2 \times 10^{16}$  ions/cm<sup>2</sup>. Inputs for  $\text{Fe}_2\text{O}_3$  were adjusted by using the “two\_comp” parameter of SDTrimSP, which can be included to adapt surface binding energies and densities from tabulated values for some pre-defined molecules. This was similarly done in [47] to achieve good agreement with irradiation experiments of Fe samples with oxide layers.

The fluence dependence of the sputtering yields clearly depicts significant preferential sputtering of O (Fig. 6a). The sputtering yield of O decreases, while the Fe yield increases and an equilibrium is reached

after around  $1 \times 10^{16}$  ions/cm<sup>2</sup>. It can be clearly seen that the sputtering yield ratios develop to the same value as the bulk concentrations of 2:3 for Fe:O. Over the simulated fluence, around 140 Å are sputtered in total (Fig. 6b). The surface concentrations (Fig. 6c), relative atomic concentration of the top 5 Å also clearly depict the preferential sputtering effects, which lead to an Fe-enrichment at the surface. At the end of the simulation, the final sample composition over depth can be plotted (Fig. 6d). The Fe-enrichment at the surface is observable as well as a layer of increased O content, where O atoms are pushed towards larger depths as a result of the collision cascade in the target.

#### 4. Conclusions

In this manuscript, a graphical user interface for the simulation program SDTrimSP was presented. This companion Python program simplifies setting up simulations with SDTrimSP, executing them and monitoring their results. Various options for input parameters as well as result evaluations are provided for both static and dynamic SDTrimSP simulations. As it was shown here, different cases for static and dynamic simulations can easily be set up and evaluated with the GUI. Ideally, this new program will become a useful tool to make SDTrimSP more accessible for simulating ion-induced effects in nuclear fusion research, planetary science, semiconductor research or other fields, where ion–solid interaction is of interest. Furthermore, it may also be of use for the education of students to learn about ion–solid interaction and its various backgrounds by visualizing relevant effects with SDTrimSP and the GUI presented in this article.

#### CRedit authorship contribution statement

**P.S. Szabo:** Conceptualization, Software, Validation, Writing – original draft, Writing – review & editing, Visualization, Project administration. **D. Weichselbaum:** Software, Validation, Writing – review & editing. **H. Biber:** Software, Validation, Writing – review & editing. **C. Cupak:** Software, Validation, Writing – review & editing. **A. Mutzke:** Software, Validation, Writing – review & editing. **R.A. Wilhelm:** Conceptualization, Supervision, Writing – review & editing. **F. Aumayr:** Conceptualization, Writing – review & editing, Supervision, Project administration, Funding acquisition.

#### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

#### Acknowledgments

This work has been carried out within the framework of the EUROfusion Consortium, funded by the European Union via the Euratom Research and Training Programme (Grant Agreement No. 633053 and Grant Agreement No 101052200 — EUROfusion). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the European Commission can be held responsible for them.

Financial support has been provided by KKKÖ (Commission for the Coordination of Fusion research in Austria at the Austrian Academy of Sciences - ÖAW) and by the Austrian Science Fund FWF (Projects No. I 4101-N36 and No. Y1174-N36). The authors are grateful to Richard van Nieuwenhoven (IAP, TU Wien) for his continued support in the software publishing process.

#### Appendix A. Supplementary data

Supplementary material related to this article can be found online at <https://doi.org/10.1016/j.nimb.2022.04.008>.

#### References

- [1] Y. Zhang, Z. Liu, M.J. Tadjer, M. Sun, D. Piedra, C. Hatem, T.J. Anderson, L.E. Luna, A. Nath, A.D. Koehler, et al., Vertical GaN junction barrier Schottky rectifiers by selective ion implantation, *IEEE Electron Device Lett.* 38 (8) (2017) 1097–1100.
- [2] W. Li, X. Zhan, X. Song, S. Si, R. Chen, J. Liu, Z. Wang, J. He, X. Xiao, A review of recent applications of ion beam techniques on nanomaterial surface modification: Design of nanostructures and energy harvesting, *Small* 15 (31) (2019) 1901820.
- [3] G. Was, R.S. Averback, Radiation damage using ion beams, in: *Comprehensive Nuclear Materials*, Elsevier Ltd, 2012, pp. 195–221.
- [4] R.E. Stoller, M.B. Toloczko, G.S. Was, A.G. Certain, S. Dwaraknath, F.A. Garner, On the use of SRIM for computing radiation damage exposure, *Nucl. Instrum. Methods Phys. Res. B* 310 (2013) 75–80.
- [5] A. Mohammadi, S. Hamidi, M.A. Asadabad, The use of the SRIM code for calculation of radiation damage induced by neutrons, *Nucl. Instrum. Methods Phys. Res. B* 412 (2017) 19–27.
- [6] S. Brezinsek, J. Coenen, T. Schwarz-Selinger, K. Schmid, A. Kirschner, A. Hakola, F.L. Tabarés, H.J. van der Meiden, M.-L. Mayoral, M. Reinhart, et al., Plasma-wall interaction studies within the EUROfusion consortium: progress on plasma-facing components development and qualification, *Nucl. Fusion* 57 (11) (2017) 116041.
- [7] A. Kirschner, D. Tskhakaya, S. Brezinsek, D. Borodin, J. Romazanov, R. Ding, A. Eksaeva, C. Linsmeier, Modelling of plasma-wall interaction and impurity transport in fusion devices and prompt deposition of tungsten as application, *Plasma Phys. Control. Fusion* 60 (1) (2017) 014041.
- [8] R. Arredondo, M. Balden, A. Mutzke, U. von Toussaint, S. Elgeti, T. Höschen, K. Schlueter, M. Mayer, M. Oberkofler, W. Jacob, Impact of surface enrichment and morphology on sputtering of EUROFER by deuterium, *Nucl. Mater. Energy* (2020) 100749.
- [9] R. Doerner, M. Baldwin, P. Stangeby, An equilibrium model for tungsten fuzz in an eroding plasma environment, *Nucl. Fusion* 51 (4) (2011) 043001.
- [10] P. Wurz, U. Rohner, J.A. Whitby, C. Kolb, H. Lammer, P. Dobnikar, J.A. Martín-Fernández, The lunar exosphere: The sputtering contribution, *Icarus* 191 (2) (2007) 486–496.
- [11] C.M. Pieters, S.K. Noble, Space weathering on airless bodies, *J. Geophys. Res.: Planets* 121 (10) (2016) 1865–1884.
- [12] H. Hofsäuss, K. Zhang, Surfactant sputtering, *Appl. Phys. A* 92 (3) (2008) 517–524.
- [13] R.M. Langford, P.M. Nellen, J. Gierak, Y. Fu, Focused ion beam micro- and nanoengineering, *MRS Bull.* 32 (5) (2007) 417–423.
- [14] J.E. Greene, Tracing the recorded history of thin-film sputter deposition: From the 1800s to 2017, *J. Vac. Sci. Technol. A* 35 (5) (2017) 05C204.
- [15] S. Dolai, R. Dey, S. Das, S. Hussain, R. Bhar, A. Pal, Cupric oxide (CuO) thin films prepared by reactive dc magnetron sputtering technique for photovoltaic application, *J. Alloys Compd.* 724 (2017) 456–464.
- [16] J. Wang, Z. Jiang, H. Chen, J. Li, J. Yin, J. Wang, T. He, P. Yan, S. Ruan, High energy soliton pulse generation by a magnetron-sputtering-deposition-grown MoTe<sub>2</sub> saturable absorber, *Photonics Res.* 6 (6) (2018) 535–541.
- [17] Y.S. Kim, H.J. Park, S.C. Mun, E. Jumaev, S.H. Hong, G. Song, J.T. Kim, Y.K. Park, K.S. Kim, S.I. Jeong, et al., Investigation of structure and mechanical properties of TiZrHfNiCuCo high entropy alloy thin films synthesized by magnetron sputtering, *J. Alloys Compd.* 797 (2019) 834–841.
- [18] A. Mutzke, R. Schneider, W. Eckstein, R. Dohmen, K. Schmid, U. von Toussaint, G. Badelow, SDTrimSP version 6.00, IPP-Report, 2019.
- [19] M.T. Robinson, I.M. Torrens, Computer simulation of atomic-displacement cascades in solids in the binary-collision approximation, *Phys. Rev. B* 9 (12) (1974) 5008.
- [20] R. Behrisch, W. Eckstein, *Sputtering by Particle Bombardment: Experiments and Computer Calculations from Threshold to MeV Energies*, Springer Science & Business Media, 2007.
- [21] A. Mutzke, W. Eckstein, Ion fluence dependence of the Si sputtering yield by noble gas ion bombardment, *Nucl. Instrum. Methods Phys. Res. B* 266 (6) (2008) 872–876.
- [22] U. von Toussaint, A. Mutzke, K. Sugiyama, T. Schwarz-Selinger, Simulation of coupled sputter-diffusion effects, *Phys. Scr.* 2016 (T167) (2016) 014023.
- [23] J.F. Ziegler, M.D. Ziegler, J.P. Biersack, SRIM—the stopping and range of ions in matter (2010), *Nucl. Instrum. Methods Phys. Res. B* 268 (11) (2010) 1818–1823.
- [24] H. Hofsäuss, K. Zhang, A. Mutzke, Simulation of ion beam sputtering with SDTrimSP, TRIDYN and SRIM, *Appl. Surf. Sci.* 310 (2014) 134–141.
- [25] P.S. Szabo, H. Biber, N. Jäggi, M. Brenner, D. Weichselbaum, A. Niggas, R. Stadlmayr, D. Primetzhofer, A. Nennung, A. Mutzke, Dynamic potential sputtering of lunar analog material by solar wind ions, *Astrophys. J.* 891 (1) (2020) 100.
- [26] P.S. Szabo, H. Biber, N. Jäggi, M. Wappl, R. Stadlmayr, D. Primetzhofer, A. Nennung, A. Mutzke, J. Fleig, K. Mezger, H. Lammer, A. Galli, P. Wurz, F. Aumayr, Experimental insights into space weathering of phobos: Laboratory investigation of sputtering by atomic and molecular planetary ions, *J. Geophys. Res.: Planets* 125 (12) (2020) e2020JE006583.

- [27] K. Arstila, J. Julin, M. Laitinen, J. Aalto, T. Konu, S. Kärkkäinen, S. Rahkonen, M. Raunio, J. Itkonen, J.-P. Santanen, et al., Potku–New analysis software for heavy ion elastic recoil detection analysis, *Nucl. Instrum. Methods Phys. Res. B* 331 (2014) 34–41.
- [28] W. Wilson, L. Haggmark, J. Biersack, Calculations of nuclear stopping, ranges, and straggling in the low-energy region, *Phys. Rev. B* 15 (5) (1977) 2458.
- [29] J. Biersack, J. Ziegler, Refined universal potentials in atomic collisions, *Nucl. Instrum. Methods Phys. Res.* 194 (1–3) (1982) 93–100.
- [30] J. Lindhard, M. Scharff, Energy dissipation by ions in the kev region, *Phys. Rev.* 124 (1961) 128–130.
- [31] O.S. Oen, M.T. Robinson, Computer studies of the reflection of light ions from solids, *Nucl. Instrum. Methods* 132 (1976) 647–653.
- [32] O. Oen, D. Holmes, M.T. Robinson, Ranges of energetic atoms in solids, *J. Appl. Phys.* 34 (2) (1963) 302–312.
- [33] J. Beeler Jr., D. Besco, Range and damage effects of tunnel trajectories in a wurtzite structure, *J. Appl. Phys.* 34 (9) (1963) 2873–2878.
- [34] M.T. Robinson, O. Oen, The channeling of energetic atoms in crystal lattices, *Appl. Phys. Lett.* 2 (2) (1963) 30–32.
- [35] O. Firsov, A qualitative interpretation of the mean electron excitation energy in atomic collisions, *Zhur. Eksptl'. I Teoret. Fiz.* 36 (1959).
- [36] J.P. Biersack, L. Haggmark, A Monte Carlo computer program for the transport of energetic ions in amorphous targets, *Nucl. Instrum. Methods* 174 (1–2) (1980) 257–269.
- [37] J. Biersack, W. Eckstein, Sputtering studies with the Monte Carlo program TRIM. SP, *Appl. Phys. A* 34 (2) (1984) 73–94.
- [38] W. Möller, W. Eckstein, Tridyn — A TRIM simulation code including dynamic composition changes, *Nucl. Instrum. Methods Phys. Res. B* 2 (1–3) (1984) 814–818.
- [39] W. Möller, W. Eckstein, J. Biersack, Tridyn-binary collision simulation of atomic collisions and dynamic composition changes in solids, *Comput. Phys. Comm.* 51 (3) (1988) 355–368.
- [40] W. Eckstein, R. Dohmen, A. Mutzke, R. Schneider, SDTrimSP, Ipp-report, 2007.
- [41] A. Mutzke, R. Schneider, W. Eckstein, R. Dohmen, SDTrimSP version 5.00, IPP-Report, 2011.
- [42] W. Möller, TRI3DYN—collisional computer simulation of the dynamic evolution of 3-dimensional nanostructures under ion irradiation, *Nucl. Instrum. Methods Phys. Res. B* 322 (2014) 23–33.
- [43] I. Bizyukov, A. Mutzke, R. Schneider, A.M. Gigler, K. Krieger, Morphology and changes of elemental surface composition of tungsten bombarded with carbon ions, *Nucl. Instrum. Methods Phys. Res. B* 266 (9) (2008) 1979–1986.
- [44] U. von Toussaint, A. Mutzke, A. Manhard, Sputtering of rough surfaces: a 3D simulation study, *Phys. Scr. (T170)* (2017) 014056.
- [45] P.G. Lucey, Mineral maps of the Moon, *Geophys. Res. Lett.* 31 (8) (2004).
- [46] W. Eckstein, R. Preuss, New fit formulae for the sputtering yield, *J. Nucl. Mater.* 320 (3) (2003) 209–213.
- [47] R. Stadlmayr, P.S. Szabo, B.M. Berger, C. Cupak, R. Chiba, D. Blöch, D. Mayer, B. Stechauner, M. Sauer, A. Foelske-Schmitz, M. Oberkofler, T. Schwarz-Selinger, A. Mutzke, F. Aumayr, Fluence dependent changes of surface morphology and sputtering yield of iron: comparison of experiments with SDTrimSP-2D, *Nucl. Instrum. Methods Phys. Res. B* 430 (2018) 42–46.