The Journal of Open Source Software

# RustBCA: A High-Performance Binary-Collision-Approximation Code for Ion-Material Interactions

**Jon. T Drobny**[1] **and Davide Curreli**[1]

**1** Department of Nuclear, Plasma, and Radiological Engineering, University of Illinois at Urbana-Champaign

**DOI:** 10.21105/joss.03298

**Software**

- Review
- Repository
- Archive

**Editor:** Arfon Smith

**Reviewers:**

- @HaoZeke
- @joglekara

**Submitted:** 08 March 2021
**Published:** 21 May 2021



## Summary

Ion-material interactions are of vital importance in industrial applications, the study and design of nuclear fusion devices, the engineering of survivable spacecraft components, and more. In particular, plasma-material interactions are typically dominated by ion-material interactions, including the phenomena of sputtering, reflection, and implantation. These phenomena are difficult to model analytically, and many such models rely on empirical or semi-empirical formulas, such as the Yamamura sputtering yield formula (Y. Yamamura, 1982), or the Thomas reflection coefficient (Thomas et al., 1992). However, such models are inherently limited, and of little applicability to complex geometry, multi-component surfaces, or for coupling to plasma or material dynamics codes. Since ion-material interactions span a range of energies from sub-eV to GeV and beyond, n-body approaches such as molecular dynamics can be computationally infeasible for many applications where the characteristic ion range exceeds the limits of reasonable molecular dynamics domains. Instead, approximations to the full n-body problem are used; the most common of these is the Binary Collision Approximation (BCA), a set of simplifying assumptions to the full n-body problem. RustBCA is a high-performance, general purpose, ion-material interactions BCA code, built for scientific flexibility and ease of use. RustBCA features include:

- electronic stopping formulations for low energy (up to 25 keV/nucleon) and high energy (up to 1 GeV/nucleon)
- Kr-C, ZBL, Moliere, and Lenz-Jensen screened coulomb interatomic potentials
- Lennard-Jones and Morse attractive-repulsive potentials
- the unique capability of using multiple interatomic potentials in a single simulation
- choice of Gaussian quadrature or the approximate MAGIC algorithm for determining scattering angles
- full trajectory tracking of ions and material atoms, including local nuclear and electronic energy losses
- a human- and machine-readable configuration file
- full 6D output of all particles that leave the simulation (via sputtering or reflection)
- multiple geometry types

## Binary Collision Approximation Codes

RustBCA is an amorphous-material BCA code, following the TRIM (Biersack & Haggmark, 1980) family of codes, which includes Tridyn (Möller et al., 1988), SDTrimSP (Mutzke et al.,

n.d.), F-TRIDYN (Drobny et al., 2017), and SRIM (Ziegler et al., 2010); this has historically been the most popular implementation of the BCA. Based on the number of citations recorded in Google Scholar at the time of writing, SRIM is the most popular amorphous-material BCA code, likely due to its being free to download, available on Windows, and having a graphical user interface. It is followed by the original TRIM code, upon which SRIM was based, then Tridyn, and finally SDTrimSP. Crystalline-material BCA codes have also been developed, such as MARLOWE (Robinson & Torrens, 1974), OKSANA (Shulga, 1984), and some versions of ACAT (Yasunori Yamamura & Tawara, 1996), but are not as widely used. The BCA itself is a set of simplifying assumptions for the ion-material interaction problem; the assumptions used in the amorphous-material BCA can be summarized as follows:

- Particles in the code, ions and material atoms both, are "superparticles" that represent many real ions or atoms each
- Energetic particles interact with initially stationary atoms in the material through elastic, binary collisions
- Collisions occur at mean-free-path lengths, or exponentially distributed path lengths for gaseous targets
- Particle trajectories are approximated by the classical asymptotic trajectories
- Electronic interactions are separated from the nuclear, elastic interactions
- Local electronic energy losses occur at each collision
- Nonlocal electronic energy losses occur along each segment of the asymptotic trajectories
- Material atoms are mobile and transfer momentum following collisions
- Particles are stopped when their energy drops below a threshold, cutoff energy, or when they leave the simulation as sputtered or reflected/transmitted particles
- Particles that leave a surface experience reflection by or refraction through a locally planar surface binding potential
- When simulating radiation damage, only material atoms given an energy larger than the threshold displacement energy will be considered removed from their original location

For detailed summaries of the history and theory of binary collision approximation codes, see the review by Robinson (Robinson, 1994) and the text by Eckstein (Eckstein, 1991).

## Statement of Need

Ion-material interactions have been historically modeled using analytical and semi-empirical formulas, such as Sigmund's sputtering theory (Sigmund, 1987), the Bohdansky formula (Bohdansky, 1984; Bohdansky et al., 1980), the Yamamura formula (Y. Yamamura, 1982, 1984; Y. Yamamura et al., 1983), and the Thomas et al. reflection coefficient (Thomas et al., 1992). However, for any physical situation beyond the regimes of validity of these formulas (e.g., non-normal angles of incidence), or for complex geometry, or for inhomogeneous composition, straightforward empirical formulas cannot be reliably used. Many BCA codes have been developed to provide computationally efficient solutions to these problems, including SRIM (Ziegler et al., 2010), Tridyn (Möller et al., 1988), F-TRIDYN (Drobny et al., 2017), SDTrimSP (Mutzke et al., n.d.) and its derivatives, which are based on the original TRIM (Biersack & Haggmark, 1980) code. However, each has limitations that prevent widespread adoption across a broad range of applications. In particular, SRIM, which is free-use but closed-source, suffers from relatively poor computational performance and significant anomalies in sputtered atom angular distributions and light ion sputtering yields (Hofsäss et al., 2014; Shulga, 2018, 2019; Wittmaack, 2004). Tridyn and F-TRIDYN, which are not open source, are limited to low ion energy, specific screened-coulomb potentials, mono-angular ion beams, atomically flat and atomically rough surfaces respectively, and are single-threaded. SDTrimSP, although significantly more advanced than the preceding codes, is built on the original TRIM source code and is not open-source.

As far as the authors are aware, there is no widely-used open-source BCA code suitable for simulating plasma-material interactions. Iradina is an open source BCA that has been used for ion-material interactions in a semicondcutor manufacturing context (Holland-Moritz et al., 2017; Johannes et al., 2014), but sputtering yields, reflection coefficients, or other key quantities of interest from iradina have not, to the knowledge of the authors, been reported for a wide range of ions, targets, energies, or angles. Additionally, those BCA codes that are available, through licensing agreements or as closed-source software, are not well suited to a wide range of physical problems. Particularly, the direct integration of BCA codes to particle or subsurface dynamics codes, such as those performed using F-TRIDYN for ITER divertor simulations (Lasa et al., 2020), requires costly external wrappers to manage simulations and process output files to perform file-based coupling. RustBCA, as part of the Plasma Surface Interactions 2 SciDAC Project suite of codes, has been developed to fill that gap and expand upon the feature set included in currently available BCA codes.

Features unique to RustBCA include the ability to handle attractive-repulsive interatomic potentials, use multiple interatomic potentials in one simulation, handle high-energy incident ions and multiple geometry types, use large file input of incident particles to facilitate coupling to other codes via HDF5, output pre-binned distributions without post-processing of text-based particle lists, and use a human- and machine-readable configuration file. RustBCA has been designed with modern programming techniques, robust error-handling, and multi-threading capability. RustBCA is being developed as both a standalone code and as a library code that may be used to add BCA routines to other high-performance codes to avoid file-based code coupling entirely. Additionally, the TRIM family of codes typically relies on the MAGIC algorithm to approximate the scattering integral with 5 fitting coefficients. RustBCA includes not only an implementation of the MAGIC algorithm, but also Mendenhall-Weller, Gauss-Mehler, and Gauss-Legendre quadrature, the three of which are significantly more accurate than the MAGIC algorithm. We hope that giving users direct access to a user-friendly, flexible, high-performance, open-source BCA will encourage and enable heretofore unexplored research in ion-materials interactions.

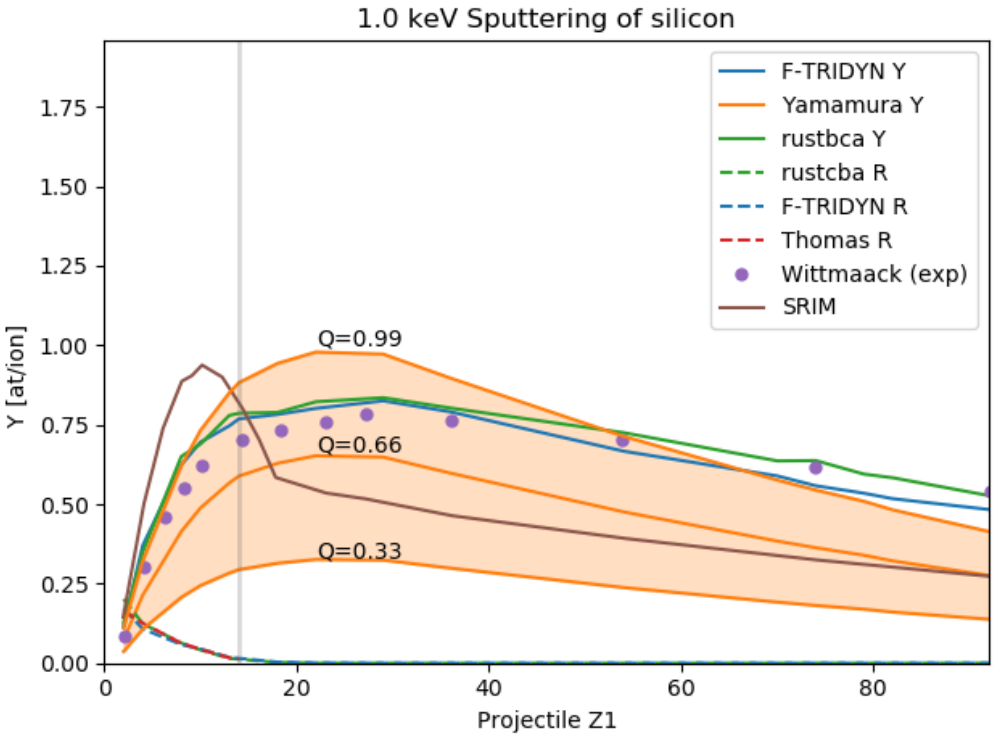


**Figure 1:** Figure showing sputtering yields of silicon from SRIM, RustBCA, F-TRIDYN, Yamamura's formula for Q=0.33-0.99, and a smooth analytical fit to experimental data by Wittmaack (Wittmaack, 2004), for an incident energy of 1 keV and for many different projectiles.



Quantities of interest from RustBCA, including sputtering yields, have been benchmarked against F-TRIDYN, SRIM, empirical formulas, and experiments. This summary figure shows the sputtering yields of silicon by 1 keV helium, beryllium, oxygen, neon, aluminum, silicon, argon, titanium, copper, krypton, xenon, ytterbium, tungsten, gold, lead and uranium ions. SRIM's unphysical Z1 dependence is clearly visible, as is the divergence of Yamamura's formula (for Q = 0.66, the reported value for silicon, and +/- 0.33) at high mass ratios (M1 » M2) from the experimental data collected by Wittmaack (Wittmaack, 2004). RustBCA and F-TRIDYN both reproduce the correct Z1 dependence of the sputtering yield, and correctly model the magnitude of the yield for all projectiles. It should be noted that, for this simulation, F-TRIDYN uses corrected MAGIC coefficients (Ziegler et al., 2010), that differ from those originally included in the Tridyn source code; Tridyn's orignal MAGIC coefficients underestimated the sputtering yield for high mass ratios. A soft grey line depicts the point of silicon on silicon sputtering. Reflection coefficients, although very low for mass ratios above one, are also shown, with F-TRIDYN and RustBCA agreeing with the semi-empirical Thomas reflection coefficient formula.

## Examples

RustBCA includes multiple example input files, under the examples/ folder on the directory, as well as discussion of each on the RustBCA Wiki page. Three examples will be summarized here.

## Example 1: Layered Targets

First, an example of 2 keV helium ions at normal incidence on a layered titanium dioxide, aluminum, and silicon target can be run in 2D with:

```
cargo run --release examples/layered_target.toml
```

The same example using the 1D layered geometry can be run with:

```
cargo run --release 1D examples/layered_target_1D.toml
```

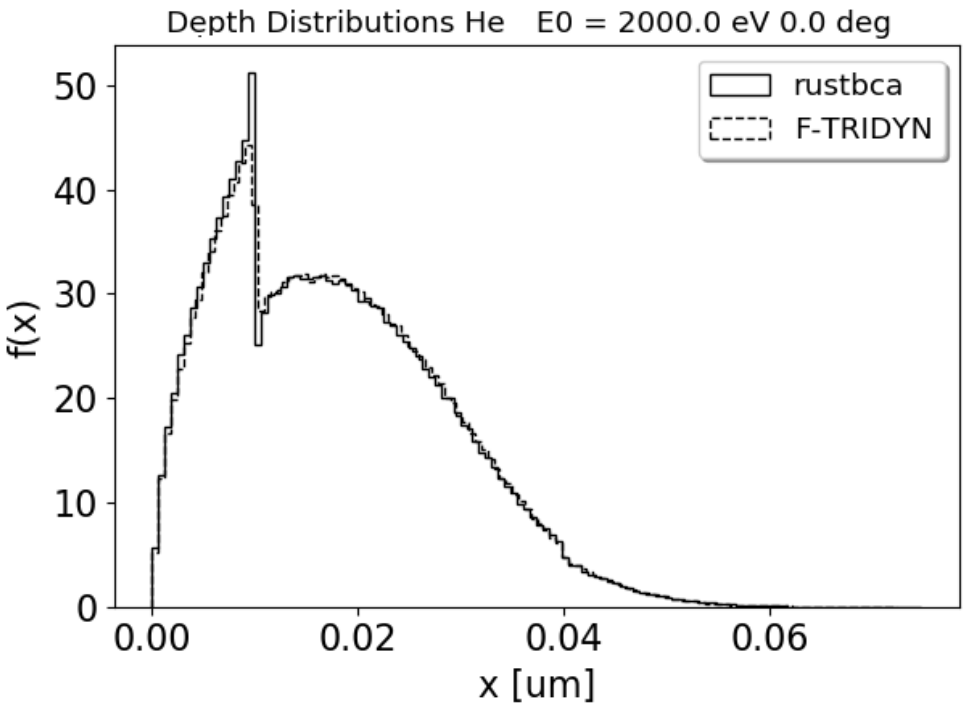


**Figure 2:** Helium implantation depth distributions at 2 keV in a layered TiO2-Al-Si target.

The depth distribution, compared to F-TRIDYN (Drobny et al., 2017), clearly shows the effect of layer composition and sharp interfaces on the combined nuclear and electronic stopping of helium.

## Example 2: 2D Geometry

Second, as an example of the capability of RustBCA to handle 2D geometry, the trajectories of 1 keV hydrogen on a circular cross-section of boron-nitride can be simulated.

```
cargo run --release examples/boron_nitride.toml
```

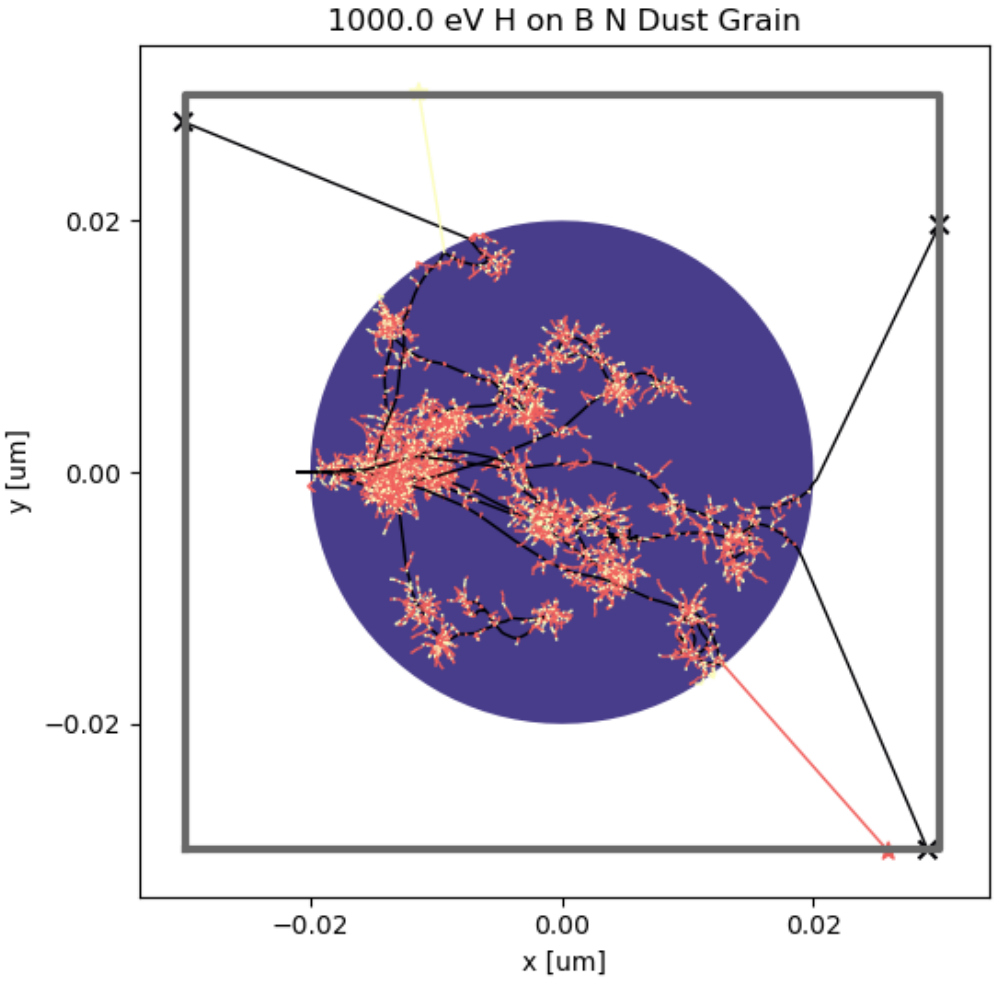


**Figure 3:** Trajectories of hydrogen and mobile boron and nitrogen resulting from 10 1 keV hydrogen ions impacting on a circular cross-section boron-nitride target.

## Example 3: Spherical geometry

Third, the 2D boron nitride example can be run as a spherical boron nitride dust grain, by running the following command:

```
cargo run --release SPHERE examples/boron_nitride_sphere.toml
```

The trajectories can be plotted in 3D with mayavi using `do_trajectory_plot_3d()` or in 2D with matplotlib using `do_trajectory_plot()` in `scripts/rustbca.py`.

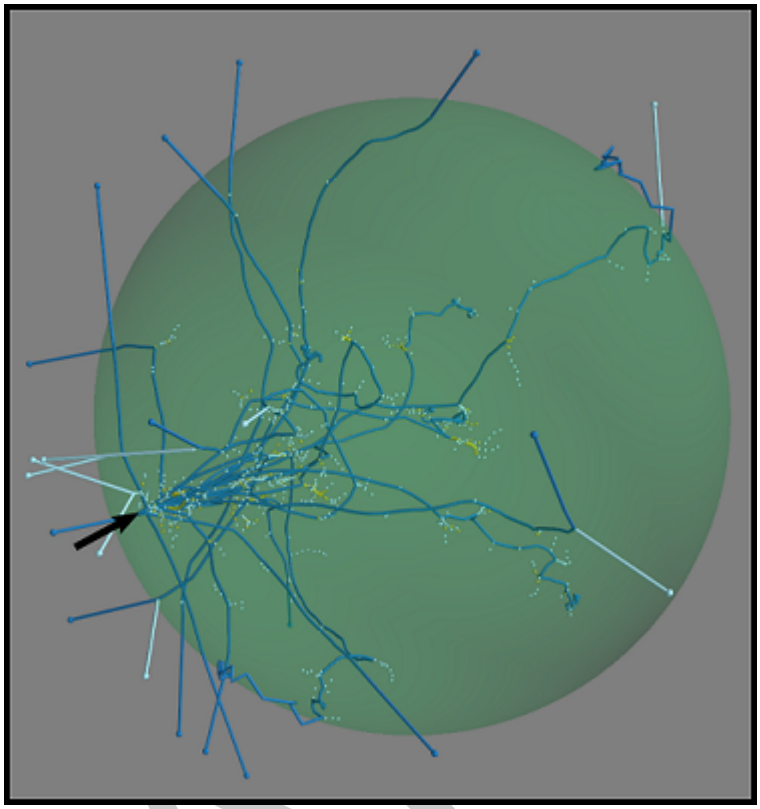

**Figure 4:** Trajectories of hydrogen, boron, and nitrogen in a 3D boron target. Hydrogen is medium blue, nitrogen yellow, and boron light blue.



## Acknowledgements

This work was funded by the U.S. Department of Energy, Office of Fusion Energy Sciences through the Scientific Discovery through Advanced Computing (SciDAC) project on Plasma Surface Interactions 2 (Grant No. DE-SC0018141).

## References

Biersack, J. P., & Haggmark, L. G. (1980). A monte carlo computer program for the transport of energetic ions in amorphous targets. *Nuclear Instruments and Methods*, *174*, 257–269. https://doi.org/10.1016/0029-554x(80)90440-1

Bohdansky, J. (1984). A universal relation for the sputtering yield of monatomic solids at normal ion incidence. *Nuclear Inst. And Methods in Physics Research, B*, *2*, 587–591. https://doi.org/10.1016/0168-583X(84)90271-4

Bohdansky, J., Roth, J., & Bay, H. L. (1980). An analytical formula and important parameters for low-energy ion sputtering. *Journal of Applied Physics*, *51*, 2861–2865. https://doi.org/10.1063/1.327954

Drobny, J., Hayes, A., Curreli, D., & Ruzic, D. N. (2017). F-TRIDYN: A binary collision approximation code for simulating ion interactions with rough surfaces. *Journal of Nuclear Materials*, *494*, 278–283. https://doi.org/10.1016/j.jnucmat.2017.07.037

Eckstein, W. (1991). *Computer simulation of ion-solid interactions*. Springer-Verlag. ISBN: 9783642735134

Hofsäss, H., Zhang, K., & Mutzke, A. (2014). Simulation of ion beam sputtering with SDTrimSP, TRIDYN and SRIM. *Applied Surface Science*, *310*, 134–141. https://doi.org/10.1016/j.apsusc.2014.03.152

Holland-Moritz, H., Ilinov, A., Djurabekova, F., Nordlund, K., & Ronning, C. (2017). Sputtering and redeposition of ion irradiated au nanoparticle arrays: Direct comparison of simulations to experiments. *New Journal of Physics*, *19*, 13023. https://doi.org/10.1088/1367-2630/aa56eb

Johannes, A., Noack, S., Paschoal, W., Kumar, S., Jacobsson, D., Pettersson, H., Samuelson, L., Dick, K. A., Martinez-Criado, G., Burghammer, M., & Ronning, C. (2014). Enhanced sputtering and incorporation of mn in implanted GaAs and ZnO nanowires. *Journal of Physics D: Applied Physics*, *47*, 394003. https://doi.org/10.1088/0022-3727/47/39/394003

Lasa, A., Canik, J. M., Blondel, S., Younkin, T. R., Curreli, D., Drobny, J., Roth, P., Cianciosa, M., Elwasif, W., Green, D. L., & Wirth, B. D. (2020). Multi-physics modeling of the long-term evolution of helium plasma exposed surfaces. *Physica Scripta*, *2020*, 014041. https://doi.org/10.1088/1402-4896/ab4c29

Möller, W., Eckstein, W., & Biersack, J. P. (1988). Tridyn-binary collision simulation of atomic collisions and dynamic composition changes in solids. *Computer Physics Communications*, *51*, 355–368. https://doi.org/10.1016/0010-4655(88)90148-8

Mutzke, A., Schneider, R., Dohmen, W., Schmid, R.;, Eckstein, W., Dohmen, R., Schmid, K., Toussaint, U. V., & Bandelow, G. (n.d.). *SDTrimSP version 6.00*.

Robinson, M. T. (1994). The binary collision approximation: Background and introduction. *Radiation Effects and Defects in Solids*, *130-131*, 3–20. https://doi.org/10.1080/10420159408219767

Robinson, M. T., & Torrens, I. M. (1974). Computer simulation of atomic-displacement cascades in solids in the binary-collision approximation. *Physical Review B*, *9*, 5008–5024. https://doi.org/10.1103/PhysRevB.9.5008

Shulga, V. I. (2018). Note on the artefacts in SRIM simulation of sputtering. *Applied Surface Science*, *439*, 456–461. https://doi.org/10.1016/j.apsusc.2018.01.039

Shulga, V. I. (2019). A comment on the computer simulation program SRIM. *Journal of Surface Investigation*, *13*, 562–565. https://doi.org/10.1134/S1027451019030339

Shulga, V. I. (1984). Computer simulation of single-crystal and polycrystal sputtering. II. *Radiation Effects*, *82*(3-4), 169–187. https://doi.org/10.1080/00337578408215770

Sigmund, P. (1987). Mechanisms and theory of physical sputtering by particle impact. *Nuclear Inst. And Methods in Physics Research, B*, *27*, 1–20. https://doi.org/10.1016/0168-583X(87)90004-8

Thomas, E. W., Janev, R. K., & Smith, J. (1992). Scaling of particle reflection coefficients. *Nuclear Inst. And Methods in Physics Research, B*, *69*, 427–436. https://doi.org/10.1016/0168-583X(92)95298-6

Wittmaack, K. (2004). Reliability of a popular simulation code for predicting sputtering yields of solids and ranges of low-energy ions. *Journal of Applied Physics*, *96*, 2632–2637. https://doi.org/10.1063/1.1776318



Wittmaack, K. (2004). Reliability of a popular simulation code for predicting sputtering yields of solids and ranges of low-energy ions. *Journal of Applied Physics*, *96*, 2632–2637. https://doi.org/10.1063/1.1776318

Yamamura, Y. (1982). Theory of sputtering and comparison to experimental data. *Nuclear Instruments and Methods*, *194*, 515–522. https://doi.org/10.1016/0029-554X(82)90575-4

Yamamura, Y. (1984). Empirical formula for angular dependence of sputtering yields. *Radiation Effects*, *80*, 57–72. https://doi.org/10.1080/00337578408222489

Yamamura, Y., Matsunami, N., & Itoh, N. (1983). THEORETICAL STUDIES ON AN EMPIRICAL FORMULA FOR SPUTTERING YIELD AT NORMAL INCIDENCE. *Radiation Effects*, *71*, 65–86. https://doi.org/10.1080/00337578308218604

Yamamura, Yasunori, & Tawara, H. (1996). Energy dependence of ion-induced sputtering yields from monatomic solids at normal incidence. *Atomic Data and Nuclear Data Tables*, *62*, 149–253. https://doi.org/10.1006/adnd.1996.0005

Ziegler, J. F., Ziegler, M. D., & Biersack, J. P. (2010). SRIM - the stopping and range of ions in matter (2010). *Nuclear Instruments and Methods in Physics Research, Section B: Beam Interactions with Materials and Atoms*, *268*. https://doi.org/10.1016/j.nimb.2010.02.091