F-TRIDYN: A MONTE-CARLO, BCA CODE FOR MODELING ION-SURFACE INTERACTIONS WITH ROUGH MATERIALS AND COUPLING PLASMA AND MATERIAL CODES

BY

JON DROBNY

THESIS

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Nuclear, Plasma, and Radiological Engineering in the Graduate College of the University of Illinois at Urbana-Champaign, 2017

Urbana, Illinois

Master's Committee:

Professor David N. Ruzic, Adviser
Assistant Professor Davide Curreli
Abstract

Fractal-TRIDYN (F-TRIDYN) is a modified version of the widely used Monte Carlo, Binary Collision Approximation code TRIDYN that has been upgraded to include an explicit model of surface roughness and additional output modes for coupling to plasma edge and material codes. Surface roughness plays an important role in ion irradiation processes such as sputtering; roughness can significantly increase the angle of maximum sputtering and strongly affect the sputtering yield. The complete effect of surface roughness on sputtering and other ion irradiation phenomena is not completely understood. Many rough surfaces can be consistently and realistically modeled with fractals, using the fractal dimension and fractal length scale as the sole input parameters to control surface morphology and roughness. F-TRIDYN includes a robust fractal surface algorithm that is more computationally efficient than those in previous fractal codes and which reproduces available experimental sputtering data from rough surfaces. Fractals provide a compelling path toward a complete and concise understanding of the effect that surface geometry plays on the behavior of plasma-facing materials. Fractals also have simple relationships to other models of surface roughness, including RMS roughness. This link has been studied and used to approximate fractal surfaces with normally-distributed statistical surfaces, which may prove more computationally efficient than explicit fractal surfaces. F-TRIDYN is a flexible code for simulating ion-solid interactions and coupling to plasma and material codes for multiscale modeling. Results from F-TRIDYN show that at high angles of incidence, sputtering yields for fusion-relevant materials increase significantly; for the case of Argon on Tungsten this is an increase of a factor of 3.5. This may effect fusion reactor performance by releasing more impurity atoms into the plasma and increasing Bremsstrahlung radiation losses. Additionally, the effect of surface morphology is significant only for ions whose mean free path in the target is on the same order as or less than the characteristic scale of the surface. Higher energy ions will experience fewer collisions at depths where the surface morphology is relevant. Code coupling has been a major focus of F-TRIDYN development, and recent results of code-coupling to a
variety of codes, including material and plasma codes, have been achieved. These preliminary results indicate that the existence of Helium impurities in a Tungsten target will increase the retention of further implanted Helium ions.
Acknowledgements

This work was funded by the US Department of Energy project on Plasma Surface Interactions - Bridging from the Surface to the Micron Frontier through Leadership Computing, through Grant No. DE-SC0004736.
# Table of Contents

List of Figures .................................................. v

1 Introduction .................................................... 1

2 Plasma-Material Interactions ................................. 4
  2.1 Introduction to Plasma-Material Interactions .......... 4
  2.2 Sputtering .................................................. 4
    2.2.1 Sputtering Yield Dependence on Incident Energy ... 7
    2.2.2 Sputtering Yield Dependence on Angle of Incidence . 7
    2.2.3 Sputtering Yield Formulae ............................ 9
  2.3 Modeling Ion-Material Interactions ...................... 14
  2.4 Binary Collision Approximation .......................... 15
  2.5 Classical Scattering ..................................... 17
  2.6 Pair Potentials for BCA codes ............................ 18
  2.7 Electronic Interactions in BCA Codes .................... 19
  2.8 BCA codes ................................................ 21
    2.8.1 TRIM and SRIM ........................................ 21
    2.8.2 MARLOWE ............................................. 22
    2.8.3 FTRIM and VFTRIM .................................... 22
    2.8.4 TRIDYN .............................................. 22
    2.8.5 SDTRIMSP ............................................ 23
    2.8.6 F-TRIDYN ............................................ 23

3 Surface Roughness in Plasma-Material Interactions .... 24
  3.1 Effect on PMI .............................................. 24
  3.2 Computational Surface Models ............................. 25
    3.2.1 "Ripples" Model ..................................... 25
    3.2.2 Local Angle-of-Incidence Distribution Model .... 25
    3.2.3 Tessellated Model ................................... 26
    3.2.4 Fractal Surface Model ............................... 26
  3.3 Modeling Surface Evolution ............................... 26

4 Fractals ....................................................... 28
  4.1 Fractal Theory ............................................ 28
  4.2 Fractal Dimension ........................................ 30
  4.3 Measurement of Fractal Dimension ....................... 33
  4.4 Fractal Generator Algorithms ............................ 34
  4.5 Fractal Generators ....................................... 36
  4.6 Fractal Surface Algorithms ............................... 37
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.7 Statistical Properties of Generated Fractals</td>
<td>42</td>
</tr>
<tr>
<td>4.8 Link to RMS Surface Roughness</td>
<td>44</td>
</tr>
<tr>
<td>5 F-TRIDYN</td>
<td>57</td>
</tr>
<tr>
<td>5.1 F-TRIDYN Code and Libraries</td>
<td>57</td>
</tr>
<tr>
<td>5.2 F-TRIDYN Output Options</td>
<td>59</td>
</tr>
<tr>
<td>5.3 Sputtering Yield Simulations</td>
<td>61</td>
</tr>
<tr>
<td>5.3.1 Helium on Beryllium</td>
<td>63</td>
</tr>
<tr>
<td>5.3.2 Deuterium on Beryllium</td>
<td>68</td>
</tr>
<tr>
<td>5.3.3 Argon on Tungsten</td>
<td>74</td>
</tr>
<tr>
<td>5.4 Surface Roughness Evolution in F-TRIDYN</td>
<td>79</td>
</tr>
<tr>
<td>5.5 Code Coupling</td>
<td>85</td>
</tr>
<tr>
<td>6 Conclusions</td>
<td>88</td>
</tr>
<tr>
<td>7 Future Work</td>
<td>89</td>
</tr>
<tr>
<td>References</td>
<td>91</td>
</tr>
</tbody>
</table>
List of Figures

1.1 An illustrated depiction of the physical processes modeled in F-TRIDYN. These include reflection, sputtering, surface morphology, damage, mixing, implantation, and layered composition. The two colors, blue and orange, represent two materials whose atoms are mixed by ion-atom and atom-atom collisions. 

2.1 A generic log-log sputtering yield vs incident ion energy curve. 

2.2 A generic curve demonstrating the dependence of sputtering yield on angle of incidence. 

4.1 Three examples of natural fractals that occur in the natural world. (a) Shows tungsten fuzz, a natural fractal created via Helium ion irradiation of Tungsten. (b) shows a member of the family Brassicaceae. (c) shows atomic-scale graphitized carbon. 

4.2 An illustration of the first seven iterations of the Middle-Third Cantor Set. 

4.3 The log-log fractal scaling law plot of a rasterized Koch Snowflake. The slope of the log-log fractal scaling law plot is shown. Inset is the rasterized Koch Snowflake used and its true, theoretical fractal dimension calculated using Eq. 4.2. 

4.4 Log-log fractal scaling law plots for several coastlines and a simple circle. Both the linear behavior of the fractal coastlines and the non-linear behavior of the non-fractal circle are readily apparent. 

4.5 Fractal generators and their respective fractal dimensions calculated using Eq. 4.2. 

4.6 A fractal generator, its encoding as a signed binary array, and a resulting fractal surface from 6 iterations using the fractal generator method. 

4.7 Three examples of generators investigated for use in F-TRIDYN. (a) is the bounded triangular generator. (b) is the unbounded trapezoidal generator. (c) is the bounded trapezoidal generator. 

4.8 A ray is cast upwards from the point P through an approximate fractal surface and the intersections that the ray makes with the surface are counted and labeled. This point is inside the surface, so the ray cast upwards from it (and indeed, any ray cast from it) will intersect with the surface 9 times.
4.9 A demonstration of the even-odd algorithm based on the Jordan Curve Theorem. In this demonstration, red Xs mark points, chosen at random, detected to be outside the surface. Blue Os mark points inside the surface. ....... 52
4.10 A demonstration of PKA reconstruction using the stepping algorithm technique. Shown in the figure are a generated fractal, a randomized copy of that fractal, used to test the method, and the resulting reconstruction of the original fractal down to the smallest ruler length used. This method accurately measures fractal dimension to 3 significant figures, as accurate as the box counting method demonstrated in Fig. 4.3 53
4.11 A histogram showing the height distribution of points of a generated fractal surface. The distribution of heights is related to the probability of a point at any depth being inside or outside of the surface. ................. 53
4.12 Three examples of PMFs for a generated fractal, showing the probability of being inside the surface versus depth. (a) shows the PMF of a first-generation fractal. (b) shows the PMF of a second generation fractal. (c) shows the complex PMF of a sixth-generation fractal. ......................... 54
4.13 An illustration of measuring RMS roughness of a surface around a mean height, with $y_i$s represented by $r_1$ through $r_7$. .......................... 54
4.14 A plot showing the proportionality constant $a$ from Eq. 4.13 for the three fractal generators in Fig. 4.7, along with a line showing how the RMS roughness scales versus fractal dimension. The RMS roughness axis, the y-axis, has been squared to show the linear dependence between the square-root of fractal dimension, $D$, and the RMS roughness itself. 55
4.15 An illustration showing a fractal surface, left, and a representation of its analogous statistical surface, right. ........ 56

5.1 The angular distributions of sputtered atoms from an F-TRIDYN simulation in x, y, and z directional cosines. ........ 62
5.2 An example implantation profile from F-TRIDYN, of 200 eV Helium ions incident upon a pure Tungsten target. ........ 62
5.3 A plot comparing F-TRIDYN results of rough and smooth surfaces to experiments and simulation results using the alternative local angle-of-incidence model of surface roughness from[25]. F-TRIDYN results are shown for a smooth (fractal dimension=2.0) surface and a rough surface whose fractal dimension was chosen to match the RMS roughness observed in the experiment plotted. .................. 66
5.4 A plot showing the effect of increasing surface roughness on sputtering yields of Helium on Beryllium at 500 eV for normal incidence and 60 degrees. ................. 67
5.5 A plot showing the log-log dependence on energy of the sputtering yield of Helium on Beryllium for three different surface roughnesses. ................................................................. 69
5.6 A plot showing the dependence of the sputtering yield on angle of incidence for a smooth surface (D=2.0) and a rough surface (D=2.1). ................................................................. 70
5.7 A plot showing the effect of increasing surface roughness on sputtering yields of Deuterium on Beryllium at 500 eV for normal incidence and 60 degrees. ................................. 72
5.8 A plot showing the log-log dependence on energy of the sputtering yield of Deuterium on Beryllium at normal incidence for three different surface roughnesses. ......................... 73
5.9 A plot showing the dependence of the sputtering yield of Deuterium on Beryllium at 500 eV on angle of incidence for a smooth surface (D=2.0) and a rough surface (D=2.1). ....... 75
5.10 A plot showing the effect of increasing surface roughness on sputtering yields of Argon on Tungsten at 500 eV for normal incidence and 60 degrees. .............................................. 77
5.11 A plot showing the log-log dependence on energy of the sputtering yield of Argon on Tungsten at normal incidence for three different surface roughnesses. ............................... 78
5.12 A plot showing the dependence of the sputtering yield of Argon on Tungsten at 500 eV on angle of incidence for a smooth surface (D=2.0) and a rough surface (D=2.1) .................. 80
5.13 A plot showing how surface roughness of relatively smooth Tungsten targets changes with incident Beryllium ions. Each line represents a set of 20 simulations run one after another from different initial conditions of surface roughness. ...... 82
5.14 A plot showing the evolution of fractal dimension of a rough Tungsten surface after 40 subsequent simulations of 100 eV incident Beryllium ions. ......................................................... 84
5.15 A plot of the evolution of a silicon surface to steady state roughness under bombardment of 100 eV, normal incidence Argon ions. ................................................................. 84
5.16 Dynamic composition over fluence of an evolving Tungsten target with 100 eV, normal incidence Beryllium ions bombarding it. ................................................................. 87
Plasma-Material Interactions (PMI) are a complex and rich field of study. Plasmas are composed of a mixture of charged particles and neutral properties and exhibit complicated dynamics. PMI are important to fields as varied as optical engineering, material processing, nanotechnology, computer processor design, and fusion. Understanding the complex interaction between ions, electrons, neutral particles is only half the story. When a plasma comes into contact with a material, rich and detailed interactions occur. Some examples of PMI processes include plasma sheath formation, ion interactions with materials, adsorption and desorption of neutral particles, secondary electron emission (SEE), and more. Energetic ions can drive significant responses in materials, and thus their interactions with materials are scientifically and economically important. Ions may undergo reflection from, transmission through, or implantation in a material. Additionally, ions may cause chemical changes in the material, cause damage to the material’s crystal structure, and more. For this reason, ion-material interactions are among the most studied PMI. Theoretical calculations are possible for simple systems, such as a single-species ion beam interacting with a monatomic target at one angle-of-incidence, but often in a plasma environment ion-material interactions are more complicated. The interplay between plasma and material can affect both plasma and material strongly. For this reason, simulations are often used to model ion-material interactions. Fractal-TRIDYN is a Monte Carlo code for simulating ion-material interactions using the Binary Collision Approximation (BCA) and includes fractal and statistical models of surface roughness. Processes included in F-TRIDYN include reflection, sputtering, surface morphology, damage, mixing, implantation, and layered composition (see Fig. 1.1).

Each of these processes is of importance to the overall response of a material to a plasma. A primary motivation for the development of F-TRIDYN is the strong effect surface roughness and surface morphology in general have on PMI. Surface roughness can change ion-material interaction quantities such as the sputtering yield by more than a factor of 2[14]. Surface roughness can change how ion-material interaction quantities depend on
Figure 1.1: An illustrated depiction of the physical processes modeled in F-TRIDYN. These include reflection, sputtering, surface morphology, damage, mixing, implantation, and layered composition. The two colors, blue and orange, represent two materials whose atoms are mixed by ion-atom and atom-atom collisions.
experimental parameters such as angle-of-incidence and incident ion energy.

A fractal model of surface roughness is attractive because it resembles physical atomic surfaces, is simple to use for any desired surface roughness, and because fractal dimension, the fractal measure of surface roughness, is measurable experimentally in many plasma devices in-situ via gas adsorption techniques. For these reasons, a fast, robust, explicit fractal surface model and associated algorithms has been implemented in F-TRIDYN. Associated algorithms include fractal surface generation, fractal dimension measurement, fractal surface reconstruction.

A focus of F-TRIDYN development has been code-coupling. Code-coupling allows simulation codes that operate on disparate timescales, length scales, and regimes to be linked. Communication between codes can be handled externally, such as in file-based code-coupling. This allows for the development of a suite of codes, each of which is responsible for a different regime of time and length scales or physical regions, that can simulate the large scale, macroscopic effects of coupled microscopic processes.
2 Plasma-Material Interactions

2.1 Introduction to Plasma-Material Interactions
Plasma-Material Interactions (PMI) are a set of complex and rich interactions and reactions between a bulk plasma and any materials in contact with the plasma. When a plasma comes into contact with a material, it forms a plasma sheath[28]. The voltage drop across a plasma sheath is often many volts, and charged particles from the plasma passing through the sheath can be accelerated to many electron-Volts of potential energy before they impact the surface.

Plasma species include electrons, ions, and neutral particles. Some plasmas may also contain larger impurity particles, referred to collectively as dust. Each of these species will interact with surfaces in contact with the plasma in different ways. Electrons are responsible for plasma sheath formation and Secondary Electron Emission (SEE), a process wherein electrons impacting the surface cause the emission of more electrons, is often responsible for sustaining plasmas. Neutral particles can adsorb onto or desorb from a surface, and can interact kinetically or chemically with the surface. Ions are responsible for a number of scientifically and economically important processes. These include reflection, implantation, sputtering, transmission, and damage. Sputtering in particular is responsible for some of the most complex interactions between a plasma and a material, as atoms from the material can enter and interact with the plasma and vice versa.

2.2 Sputtering
Sputtering is a process by which energetic ions incident upon a material collide with and expel atoms from the material. Exiting atoms may or may not be ionized depending on the initial energy and mass of the incident ion. Sputtering was first observed via its effects as early as 1852 in gas discharge tubes as charged ions eroded the surface of cathodes[5]. In the context of plasma-material interactions, sputtering is but one of a number of processes that energetic ions (above the sputtering threshold for a given material)
may undergo. Incident ions may sputter atoms from the material, reflect or backscatter following small-impact-parameter collisions with the surface, transmit through the material, implant within the material, or react with the material[15]. Though each of these processes plays a role in PMI, sputtering is among the most important to scientific research and engineering.

Sputtering is caused by ion-atom and subsequent atom-atom collisions. A series of collisions caused by an incident energetic particle that may or may not lead to sputtered atoms is known as the collision cascade. Collision cascades develop on the picosecond time scale. As an example, at 10 keV, Copper ions incident upon a Copper target will set as many as 300 particles into motion before the slowing of the incident ion. The number of particles in motion decreases to zero in less than 1 ps[42]. These are truly atomic timescales. Timescales this small mean that energetic ion-driven processes occur faster than thermal processes such as diffusion, which occurs on the timescale of hundreds if not thousands of seconds[24]. In many situations, these processes are thus non-thermal or athermal, and can be used to induce changes in materials not possible with conventional thermal or chemical methods[34]. Understanding of processes such as sputtering lays the groundwork for the continued development of the science of targeted etching and deposition that is currently fundamental to the fields of nanoengineering and computer engineering. Targeted etching and deposition are used to create atomic scale structures such as the microscopic transistors and circuits on computer processors[28].

Sputtering is often described quantitatively with the so-called sputtering yield, measured as, on average, how many target atoms are sputtered, or ejected from the material, per incoming ion. The sputtering yield is an important factor because it sets the rate at which particles are emitted from a surface when exposed to plasma, a large part of the total recycling process. This is important for fusion plasma physics, where impurity particles play a significant role in plasma dynamics and surface processes, such as redeposition and erosion. Additionally, it is possible to approximate sputtering yields via empirical and semi-empirical formulae of various complexity, meaning that sputtering yields can be used in place of a full ion-material simulation to act as an interface between plasma and material codes in many contexts. In certain systems, however, such as the Helium-Hydrogen-Tungsten system, there may be too many variables (surface roughness, surface composition, material erosion, incident energy and angular distributions, etc.) to capture the behavior of the sputtering yield with simple formulae. In such systems, one must use a full simulation of ion-material interactions, like a BCA code.
like F-TRIDYN, in order to fully capture the interface between the plasma sheath and the material at atomic time- and lengthscales.

Sputtering is among the most scientifically and economically important ion-solid interactions. At an energy above the sputtering threshold, or the energy at which sputtering becomes significant, incident ions cause material atoms to exit via transfer of momentum due to a series of nuclear collisions. This process is of economic importance because it is the primary driver of ion etching and erosion processes in industry. Sputter etching is a process by which a material is selectively eroded away via simple sputtering. This process is unselective but effective[28], but the desired product is the sputtered atoms themselves. These atoms, once sputtered from a target material, will leave the material with some velocity based on the dynamics of the collision cascade. These atoms are often neutralized, meaning they will travel in straight lines from the target to deposit upon the substrate.

Sputtering is also of profound importance in other plasma devices such as fusion reactors. Impurities in a fusion plasma are a major cause of energy loss via Bremsstrahlung radiation. This is a process by which accelerating charged particles radiate some of their energy away. Specifically, this process results in an energy loss that depends on the charge of the atom in question. If an impurity atom in a fusion reactor has a relatively high atomic number, it will cause significant radiation energy loss due to the quadratic dependence on average atomic number in the Bremsstrahlung loss term in the fusion reactor energy balance (see Eq. 2.1)[17].

$$S_B = C_B \sum_j \frac{Z_j^2 n_j}{n_e} n_e^2 T_e^2$$

As the average atomic number of the fusion plasma increases, performance decreases quadratically. High-Z impurities in the plasma may have been sputtered from walls, the result of chemical processes at the walls, or consist of dust emitted from the wall via ion bombardment or mechanical and heat stresses. Understanding the full dynamics of sputtering systems can allow one to choose materials that have low sputtering yields and favorable material properties in order to avoid loss of fusion plasma performance in experimental reactors.

Sputtering yields depend on a multitude of factors. For a given system, that is, given species of incident ions and target atoms, the parameters that can be controlled experimentally include the energy distribution of incident ions, the angular distribution of incoming ions, and possibly the surface morphology. Sputtering yield formulae, both empirical and semi-empirical,
have been developed to give an approximate form of the dependence of sputtering yields on each of these experimental parameters.

### 2.2.1 Sputtering Yield Dependence on Incident Energy

When measured against the energy of the incident ion, the sputtering yield function for most systems can be divided into three regimes, as marked in the generic sputtering yield versus energy curve illustrated in Fig. 2.1. In the low energy, or pre-threshold, regime, the sputtering yield is effectively zero. Depending on the system in question, this regime can span from 0 eV to as high as tens of electron-volts. Dividing the first and second regimes is the sputtering threshold, which is defined qualitatively as the energy at which the sputtering yield begins to rise sharply from zero. A minimum amount of energy is required to displace a material atom from its lattice position, known as the displacement energy. Incident ions with energies below or near this displacement energy will not have enough kinetic energy to cause any target atoms to be knocked out of their lattice position, leading to near-zero sputtering. Tungsten is the most readily available material with such a relatively high sputtering threshold, measured at approximately 165 eV[20]. In the second regime, the sputtering yield rises dramatically as the total amount of energy deposited into the material increases. This continues post-threshold before reaching a maximum, where the distribution of deposited energy is too far below the surface for sputtering to be efficient. After the maximum yield, the sputtering yield decreases slowly. This is a property of high energy ions implanting too deep within the material to cause sputtering.

Fig. 2.1 shows a generic example of a log-log plot of sputtering yield versus incident ion energy. The sputtering threshold is marked as the place where the sputtering yield becomes non-zero. Left of the sputtering threshold is the pre-threshold regime. The second regime, where the sputtering yield is increasing, is marked on the figure. At maximum sputtering yield, the competing processes of increasing implantation depth and increasing energy transfer to the target are equalized. Past this point, the sputtering yield decreases as the incident ions are implanted too deep and their energy deposited far below the surface.

### 2.2.2 Sputtering Yield Dependence on Angle of Incidence

When measured against the angle of incidence, the sputtering yield function can be divided into two regimes. As the angle of incidence is increased
Figure 2.1: A generic log-log sputtering yield vs incident ion energy curve.
past normal incidence, the depth distribution of energy transferred from
the incident ion to the material will concentrate near the surface, causing
more atoms from the surface to be sputtered. Between normal incidence
and the angle of maximum yield is the first regime. As the angle is further
increased, the transfer of energy to the material decreases, and fewer atoms
receive enough kinetic energy from the collision to sputter. At perpendicular
incidence, a flat surface will obviously have a sputtering yield of zero. Be-
tween the angle of maximum yield and perpendicular incidence is the second
regime.

A generic sputtering yield versus angle curve for a smooth surface is
shown in Fig. 2.2. The two regimes are marked on this figure, showing the
upward trend of yield versus angle in the low angle-of-incidence case, de-
scribed first theoretically by Sigmund[48], and the high-angle case, wherein
the sputtering yield decreases to zero. Angles of maximum sputtering yield
for different ion-target combinations may be used in processes where sputter-
ing is desirable. These include magnetron sputtering, surface smoothing and
roughening, and erosion. It is possible to design a sputtering system with
an incident ion angular distribution that has a high enough mean angle to
normal to increase sputtering yield significantly. Smoothing and roughening
processes depend strongly on the exact crystalline structure of the target
material and are thus not easily simulated in F-TRIDYN, which assumes
an amorphous surface. Other BCA codes, such as MARLOWE and crystal-
TRIM, have been developed to include a crystal structure[36][40]. Low sput-
tering yields at high angle are utilized in current designs for divertor tiles
in fusion reactors, relying on the high angle of incidence of magnetic field
lines as they approach the divertor through the accelerating plasma sheath.
These devices utilize high-angle magnetic field lines near the surface of the
divertor tile in order to decrease sputtering of the divertor tile material[17].

2.2.3 Sputtering Yield Formulae

Developing useful formulae for calculating sputtering yields has been a task
of interest since the discovery of sputtering processes. It was not possible to
simulate ion-solid interactions on many early computers, so an alternative
strategy of developing empirical or semi-empirical, multi-variate fits to sput-
tering yield curves from experiments was pursued. Even today, as modern
ion-solid interaction simulations can be run even on a low cost laptop com-
puter, the use of sputtering yield formulae is still preferred in fields where
strong assumptions about the studied system can be made. Most sputtering
Figure 2.2: A generic curve demonstrating the dependence of sputtering yield on angle of incidence.
yield formulae assume that the target is a smooth, simple material, and that
the incident ions are monoangular and monoenergetic. For many use cases,
such as materials processing, these formulae are sufficient. For other use
cases, such as in fusion, strong assumptions cannot be made about the ions
or the material due to dynamic system properties such as surface roughness,
material composition, energy and angular distributions, and more. Simula-
tions, not formulae, must be used in whole-device models of fusion device
walls and any other scenario where the interplay between plasma and solid
is complex and the length- and timescales range from the atomic to the
macroscopic.

Sigmund's formula for sputtering yield is amongst the earliest theoretical
contributions to the science of sputtering. His main paper, focusing on the
collision cascade as a theoretical object and as the main driver of sputtering,
was published in 1969[48]. Collision cascades driven by ion bombardment
were then understood to consist of two primary mechanisms that drive the
interaction. The first of these is the slowing down period of the incident ion
and all primary recoil target atoms that obtain similar amounts of energy
from collisions. This phase is fast, as each collision transfers significant
energy and momentum from the energetic incident ion to a series of target
atoms. The amount of energy transfer per purely elastic collision depends
on the mass of the two particles that collide and the angle at which the
incident ion scatters in the center of mass frame. This energy dependence
is reproduced in Eq. 2.3[15]. In Eq. 2.2, $M_1$ and $M_2$ refer to the masses
of the two colliding particles and $E_r$ refers to the relative, center-of-mass
energy of the system. This form implies a maximum energy transfer, Eq. 2.2,
separated and denoted $T_m$.

$$T_m = \frac{4}{1 + \frac{M_2}{M_1}} E_r$$  \hspace{1cm} (2.2)

$$T = T_m \left( \sin \frac{\theta^2}{2} \right)$$  \hspace{1cm} (2.3)

Derived from central force potential classical mechanics, this equation
implies that an incident ion with initial kinetic energy above this maximum
energy transfer may have to experience multiple collisions before it slows
down to thermal speeds. Sputtering thresholds are highly dependent on
the maximum energy transfer term and the displacement energy, since if an
incident ion does not have enough energy to displace a target atom even
at maximum energy transfer, the sputtering yield will be zero as seen in
the pre-threshold regime of the sputtering yield dependence on incident ion
energy.

The second phase of the collision cascade is the initialization of lower energy atomic motion and concerns the motion of the vast majority of atoms involved in the cascade\cite{48}. This phase is significantly slower than the first, as the incident ion has had the time to deposit the vast majority of its kinetic energy to the surrounding material. Sigmund recognized that the first phase of the cascade is primarily responsible for dictating the size of the cascades effects, while the second phase is primarily responsible for effects that happen on longer timescales.

Recognizing these two phases as distinct processes separated by vast differences in average atomic energies was the first step to understanding sputtering from a theoretical standpoint. As important is the recognition that almost all atoms eventually sputtered from a target during a sputtering event begin very near the surface of the target. The maximum depth from which sputtered atoms emerge from the target is approximately 5 angstroms\cite{48}. This depth is of the same order of magnitude as the interatomic separation of most materials. For example, the interatomic separation of Tungsten in its standard FCC crystal structure is approximately 3.16 Angstroms at STP\cite{51}. Essentially, for a material like tungsten, sputtered atoms emerge from only the first atomic layer of the surface. This fact means that the surface binding energy of a material plays a large role in sputtering yields and is involved strongly in theoretical and semi-empirical formulations of yield formulae.

J. Bohdansky formulated a so-called universal relationship for the sputtering yield of monatomic solids at normal incidence\cite{10}. The so-called Bohdansky formula is often used as a first-step approximation to sputtering yield calculations in place of a more complicated simulation of sputtering via ion-irradiation. Bohdanskys primary goal was the investigation of the effect of impurity emission via sputtering from the first wall of a fusion reactor. Impurity concentration is the cause of one of the primary energy loss sources in a fusion reactor, Bremmstrahlung radiation. Bremmstrahlung radiation losses scale quadratically with the average atomic number (and therefore charge) of the fusion plasma in a tokamak. If there is a significant quantity of high-Z impurity atoms in the plasma, energy loss via Bremmstrahlung radiation will be significant. The Bremmstrahlung energy loss term for a fusion power reactor is detailed in Eq. 2.1. Bohdanskys relation is based on a modification of Sigmunds sputtering theory to include linear cascade theory for light ions, the case of interest for fusion fuel gas sputtering of first wall tiles. Bohdanskys formula is reproduced in Eq. 2.4\cite{10}. In the
Bohdansky relation, $\alpha$ is a parameter that depends only on the mass ratio and is essentially the energy-independent parameter of the relation. $S_n$ is the energy-dependent term and $E_{th}$ is the sputtering threshold. $R_p$ is the range of the ion and $E$ is the incident energy. $U_0$ is the surface binding energy of the target. This relation is only valid for light-ion sputtering.

$$Y(E) = \frac{0.042}{U_0} (R_p/R) \alpha(M_2/M_1) S_n(E) \left[ 1 - \left( \frac{E_{th}}{E} \right)^{2/3} \right] \left[ 1 - \left( \frac{E_{th}}{E} \right)^2 \right] (2.4)$$

The Bohdansky relations main improvement upon the Sigmund theory was the inclusion of inelastic energy losses via a free parameter and was found to agree with experimental data for more than 100 ion-atom combinations\[10\]. Because incident ions lose energy to both elastic nuclear and inelastic electronic interactions, the inclusion of the inelastic term in Bohdanskys formula significantly improved its physical accuracy over previous sputtering yield formulae.

Before the advent of effective computer simulation of ion-solid interactions, sputtering theories were mostly restricted to theoretical investigation and empirical and semi-empirical fits to experimental sputtering yield curves. Early experimental investigations into the dependence of the sputtering yield on angle of incidence found an approximate $\cos^{-1} \theta$ dependence for low angles of incidence, and early theoretical work by Sigmund predicted a $\cos^{-f} \theta$ (where $1 < f < 2$) dependence for the normalized sputtering yield for the same regime\[52\]. Past the regime of near-normal angles, at energies significantly greater than the sputtering threshold the angular dependence for the normalized sputtering yield was estimated to be approximately $1.2(\theta/\theta_{max})^2$, where $\theta_{max}$ is the angle of maximum sputtering yield\[52\].

A significant downside of these early investigations was that the validity of the formulae was limited to specific regimes of angles of incidence, and no one theory was available to describe the sputtering yield of a wider range of angles of incidence. This meant that sputtering yield formulae are only useful in situations where the energy or angular distributions of incoming ions are static. Yamamura et al. developed a theory that was an attempt to describe simply the angular dependence on sputtering yield for both a wide range of systems and angles of incidence\[52\]. These formulae depend on the angle of maximum sputtering yield, a fit-parameter $f$, the threshold energy, the reflection coefficient, the sublimation energy of the target material, the average energy of reflected ions, and the deposited near-surface energy by a backscattered ion. All of these physical parameters are readily available in
experimental databases, but often have significant variation and thus must be chosen with care.

All sputtering formulae suffer from similar weaknesses. First, they can only account for systems with a small number of independent variables. Most can only account for the species of the incident ions and the solid, the incident energy, and the incident angle. Improvements to the various formulae to include surface roughness have been made, but are not widely used[50]. Other factors, however, are more difficult to include in a sputtering yield formula. In many situations, the surface exposed to the plasma is not composed of only one species, but rather has some range of spatially-dependent composition changes. For example, in a fusion setting, a Tungsten divertor tile may have redeposited Beryllium on its surface, Helium clusters implanted below the surface, and trapped Hydrogen atoms along atomic defects[37]. Additionally, surface morphology can be quite complex, resulting in difficult-to-predict changes to average sputtering yields based on the detailed nanostructure of the material. These changes can have a drastic effect on sputtering yields and other ion-solid interaction parameters[35]. In situations where a sputtering yield formula cannot fully account for the complex, interconnected behavior of a particular plasma-material interaction, one must use a simulation method instead.

2.3 Modeling Ion-Material Interactions

Two of the most studied simulation methods for ion-solid interactions are the Binary Collisions Approximation (BCA) and Molecular Dynamics (MD)[41]. BCA codes rely on the assumption that nuclear ion-solid interactions can be treated as a series of binary collisions as the name of the approximation implies. This assumption has a limited range of validity, eV to MeV, but, if the correct interatomic potential is chosen, can reproduce experimental results such as sputtering yields at energies far below the range of validity of the binary collision assumption[15]. Molecular Dynamics codes are usually classical or pseudo-classical mechanics codes that evolve a system of hundreds of thousands to hundreds of millions of atoms that interact at each time step with each other via interatomic potentials.

Both BCA and MD codes have advantages and disadvantages. MD codes capture significantly more detailed physics than BCA codes, since every atoms position is tracked explicitly at each time step. This is also a downside, however, because MD codes are significantly slower than BCA codes. The computational complexity of MD codes scales as $N^2$ (where $N$ is the
number of particles) because each particles interactions must be handled with every other particle in the code. Capturing high energy behavior that encompasses a large system size, such as ion-irradiation, often requires a prohibitively high number of particles for any computer system that is not at High-Performance Computing (HPC) scale. Additionally, MD codes can be very sensitive to changes in interatomic potentials, so extreme caution must be used when interpreting results from MD codes if interatomic potentials between all species are not known to sufficient precision.

BCA codes, when used to study the same system, are much faster than MD codes. Their computational complexity scales linearly with the active number of particles in each collision cascade. Hence, even extremely high energy ion-irradiation events can be simulated in a BCA framework at relatively little computational expense, even as a large number of atoms are set into motion by each initial ion. BCA codes are also not particularly sensitive to choice of interatomic potential, and agreement with experiment has been found with several choices of interatomic potential\cite{21}\cite{41}. This means that so-called universal potentials, such as the Moliere potential, can be used in place of detailed knowledge of the complex quantum interactions between all participating species. On the other hand, MD codes have been found to be more accurate for some systems\cite{13}. This means that one must carefully consider the potential trade-offs between using MD codes and BCA codes when studying ion-material interactions. For code coupling purposes, requiring on-line performance, a BCA code must be used. F-TRIDYN is a Monte Carlo BCA code, taking advantage of fast algorithms for calculating ion-solid interactions according to the simplifying assumptions that make up the BCA, developed primarily for code-coupling towards whole-device modeling of fusion reactors and plasma devices.

### 2.4 Binary Collision Approximation

The Binary Collision Approximation (BCA) is a set of assumptions that simplify the simulation of energetic ions interacting with materials. The assumption of binary collisions is used in many physical problems, but the BCA in the context of plasma-material interactions refers specifically to the following assumptions:

- Energetic charged particles interact with materials in a series of discrete collisions
- Nuclear collisions are binary and occur at mean-free-path distances in
Electronic interactions are handled quasi-elastically and calculated at the apsis of each collision using either local or non-local models

Target particles are displaced only if the energy transfer during collisions is above a threshold

Particles are stopped when their kinetic energy drops below a threshold referred to as a cutoff energy

Effects of collisions are calculated by numerical integration of the classical scattering integral, usually assuming a universal screened Coulomb potential such as the ZBL, Kr-C, or Moliere potentials[15].

The first assumption is responsible for the computational efficiency of BCA simulations. Because at each collision step there are only ever two particles interacting with each other, the BCA problem scales very effectively. The computational complexity increases only linearly with the number of active particles in the simulation, which of course will depend on the incident ion energy. If the ion energy is high, the simulation will take more computer time than the same system with lower energy ions.

Nuclear collisions are restricted to binary classical scattering problems. Each collision is chosen to occur a priori at mean-free path (mfp) distances in the material based on the number density of the material. This assumption is valid only in the Monte Carlo framework and effectively ignores crystal structure and enforces the assumption of amorphous material structure. Specific effects, such as ion channeling, will thus not be seen in Monte Carlo BCA codes with this assumption, but will be seen in BCA codes that include an explicit crystal structure, such as MARLOWE[36][41].

Actual electronic interactions are a complicated quantum process. MD codes have been developed that handle these quantum interactions in a pseudo-classical way[11], but for the energies involved in typical ion-solid interaction problems a more simplified assumption is valid. By restricting the calculation of electronic stopping to the apsis of each nuclear collision, the code achieves a remarkable speed advantage over quantum electron interaction codes.

Only initiating collisions above a set threshold energy restricts the number of particles set into motion by the incident ion. If a particle is transferred an amount of energy lower than its displacement energy, it is assumed that the energy transfer caused in-place vibration of the target atom but did not
allow it to break free from its lattice site. The BCA is not particularly sensitive to the threshold energy, as described later in this work, so it is suitable to round this value to single-digit eV values based on the energy it takes to remove the target atom from its lattice position.

Stopping particles at a cutoff energy also restricts the number of particles being actively tracked by the code at a given time. This allows for particles to come to rest with some remaining thermal energy at an off-lattice position and remain an interstitial impurity in the target.

Assuming two-body interaction potentials is the same assumption made in most classical MD codes. Its validity is strongest at higher energies, where chemical effects can be neglected. Using a screened potential for low energy ion-solid interactions in BCA codes can still produce valid data due to a convenient and coincidental cancellation of errors.

Nuclear collisions are handled via a numerical integration of the classical scattering integral in all BCA codes. The classical scattering integral is solved via numerical quadrature to find scattering angles and magnitudes of energy transfer for each collision. BCA codes can be used for the interaction with any state of matter where binary collisions can be assumed to be the dominant form of interaction. This includes material processing, certain processes in fusion reactors, Plasma-Material Interactions generally refers to solids and liquids. It is common in BCA codes to assume that the solid can be approximated as amorphous, even when this is not the case. It is possible to include crystalline structure in a BCA code, but this is computationally more expensive than randomly picking impact parameters and azimuthal angles from distributions. Relatively few codes have attempted to include a crystalline structure. The most commonly cited code that includes a structure is MARLOWE[36][40].

2.5 Classical Scattering

Scattering refers to a class of problems where two massive particles interact via a central force. Classical scattering problems are solved via the so-called scattering integral, which provides all relevant quantities of interest to the problem. These quantities of interest include the scattering angle, \( \theta \), and the energy transfer during the collision. Eq. 2.5 shows the full form of the classical scattering integral, where \( r \) is the distance between the two colliding atoms and \( R \) is the distance of closest approach[5].
\[
\theta = \pi - 2p \int_{R}^{\infty} \left( r^2 / \sqrt{1 - \frac{V(r)}{E_r} - \frac{p^2}{r^2}} \right) dR
\]

(2.5)

Eq. 2.5 is the classical scattering integral for a central force potential, \(V(r)\). In order to calculate the scattering angle from the scattering integral, the relative kinetic energy in the center of mass frame, \(E_r\), the impact parameter, \(p\), the distance of closest approach, and the central force potential must be known. The impact parameter is a term that accounts for the off-axis distance of the initial trajectory of the scattering particles relative to the target. Given a specified system of species 1 and 2 with appropriate masses, scattering angles and energy transfer can be reduced to a two-dimensional problem, the two dimensions being the incident energy before the collision and the impact parameter. Since scattering integrals are relatively difficult to calculate, pre-calculation of scattering integrals for a given system of two species for a range of possible impact parameters and incident energies is possible. Pre-calculation of scattering integrals may be a useful method to increase the efficiency of BCA and other scattering codes.

In the context of nuclear collisions, the central force involved is known as the interatomic potential. Classical charged particles share a common central force potential, the Coulomb potential, characterized by an inverse dependence on distance from the particle. Atoms, however, are more complicated than simple point charges and consist of varying quantities of electrons and protons that affect the magnitude of the effective central force they exert on another charged particle through the mechanism of charge screening. A powerful method of handling the complex quantum physics that drives the detailed dynamics of this system is the screened potential method, wherein the various quantum effects on the system (including spin interaction, Thomas-Fermi statistics on the electronic system, and more) are reduced to a single, one-dimensional screening function.

### 2.6 Pair Potentials for BCA codes

Several pair potentials have been investigated thoroughly for use in BCA codes[15]. These include:

- Moliere potential
- Bohr potential
- Thomas-Fermi potential
• Lennard-Jones potential
• ZBL potential
• Kr-C potential

Moliere’s potential is a form of the Born Mayer potential with an effective-charge approximation for the screening length. This potential is essentially an approximation to the Thomas-Fermi potential. The Bohr potential is the simplest approximation, using a single exponential factor for screening\[15\]. The Thomas-Fermi potential was derived from the quantum statistics of many-electron systems\[19\]. Lennard-Jones potentials are a form of universal potential popular for simple MD codes of nonreactive gases such as Argon\[18\]. This potential has two free parameters. First, the depth of the potential well is a free parameter that can be tuned based on experimental scattering measurements. Second, the distance of the minimum of the potential is used as a second free parameter that controls the interatomic spacing of the simulated material. The ZBL potential is an RMS average of 500 combinations of ion-target systems and includes approximations for charge exchange and correlations\[55\]. ZBL potential is very often used, but there is some evidence that it is too strong at longer distances yet more accurate than the similar Moliere potential at shorter distances\[15\]. The Kr-C potential, so called because it is based on the Krypton-Carbon system, can be similar to either the ZBL or Moliere potentials depending on choice of parameter. This potential is used in TRIDYN and F-TRIDYN and a good example of a practical universal potential\[32\].

2.7 Electronic Interactions in BCA Codes

Electronic interactions in F-TRIDYN are handled quasi-elastically, based on the electronic stopping theory of Lindhard and Scharff\[29\]. This model assumes that the electronic interaction can be represented using a free electron gas model. This separates the problem of ion-solid interactions into the binary, nuclear collisions and non-elastic interactions with a free electron gas. Lindhard and Scharff used the following assumptions\[54\]:

• Electronic systems are modeled as a free electron gas comprised of plane waves at absolute zero with uniform density

• Charge neutrality is maintained with a fixed, uniform, positive charge background that represents the effects of the nuclei
- Charge interactions with the electron gas are treated as perturbations upon it
- No particles are relativistic

These assumptions greatly simplify the inherently quantum problem of electronic interactions. When compared to the energies of incident ions in sputtering scenarios, assuming that the electron gas is at absolute zero is a good assumption. The conversion between eV and degrees Celsius of a bulk system is on the order of \(10,000\text{C/eV}\). Thus, there is so great a difference between energetic ion "temperatures" and the background temperature of the target that the target temperature may be neglected in many cases. Some experiments have found an effect of material temperature on sputtering yield, but these are not handled implicitly in BCA codes[12].

Charge neutrality is necessary to prevent unwanted heating or expansion of the system, as a system composed purely of particles with repulsive forces will be unstable. Materials can accumulate some charge on their surface, but that is not taken into account for modeling the bulk electronic system in BCA codes.

Perturbation theory and the assumptions of uniform density allow results to be calculated analytically to close the problem of electronic stopping power. By making these assumptions, the mathematical complexity of the problem can be greatly reduced. Few ion-irradiation problems involve relativistic velocities. Including relativistic effects would significantly complicate the already-complex dynamics of the scattering integral, and classical scattering theory would no longer be sufficient. As long as the incident ions are not approaching the material at relativistic velocities, this assumption is valid.

Using these assumptions, it is possible to derive the double integral equation for measuring the electronic interactions in a bulk material known as the electronic interaction term. From this equation, it can be found that the maximum strength of interaction occurs when particles are traveling at the Fermi velocity in the free electron gas that is formed from the combined electronic systems of all target atoms in their respective lattice positions. Faster particles have less time to interact with the electrons in the material and slower particles collisions are adiabatic in nature and are not as strong[29].

Future work on the BCA may include updating the electronic interaction model. The Lindhard-Scharff model was developed decades ago, and many advancements in the study of many-bodied charged quantum systems have
been made since then. Increasing the accuracy of the electronic interaction could lead to a significant increase in the energy regimes where the BCA remains relevant. For example, the BCA is accurate for sputtering yields only by coincidence of cancellation of errors near the sputtering yield threshold. This implies that an improvement is possible in increasing the accuracy of the BCA either through coupling to an explicit physics solver such as an MD module or some other means of including more detailed electronic and chemical interactions. Such an advancement is in the early planning stages for F-TRIDYN and will be considered as part of future research on the BCA especially for use in code-coupling contexts for whole-device modeling of future fusion reactors.

2.8 BCA codes

Many BCA codes have been developed and used for modeling ion-material interactions. Several particular codes have historical and contemporary significance due to the prevalence of their use or their availability. These include TRIM[6] an early BCA code, MARLOWE[42], a BCA code that includes a crystalline surface, SRIM[56], based on TRIM, TRIDYN[32], a BCA code that included dynamic target composition and based on TRIM, FTRIM[45] and VFTRIM[47], two codes based on TRIM that included a fractal model of surface roughness, SDTRIMSP[33], based on the same interaction potential as TRIDYN and upgraded to include various models of surface morphology in 2 and 3 dimensions, and F-TRIDYN[14], an upgraded version of TRIDYN including models of surface roughness and additional I/O options for code-coupling.

2.8.1 TRIM and SRIM

SRIM is the most widely used BCA code for a number of reasons. First, it is simple to use. SRIM includes a graphical user interface that allows for anyone with knowledge of the system they wish to simulate to run a TRIM simulation. It is a relatively old code, based on earlier versions of TRIM, and is very well tested against experiment and other simulation methods[55], however, is not the most accurate BCA code that is available.

Several problems have been found with SRIM results that are not present in more modern BCA codes, such as TRIDYN and SDTRIMSP[21]. Angular sputtering yield curves deviate significantly from both experiment and other codes. During tests using 1 keV Xe incident on Ge and Si targets, it was found that both SRIM 2008 and SRIM 2014 displayed different and
incorrect behavior[21]. TRIDYN and SDTrimSP on the other hand matched experiment and each other. Additionally, sputtered atom angular distributions were incorrect in SRIM. Even for relatively high angles of incidence, SRIMs sputtered atom angular distributions were symmetric. These results essentially invalidate SRIMs usage for both sputtering yields and code coupling at non-normal angles of incidence. Because SRIMs code is closed source, it is not known where the problem is. Hofssäss proposes that the issue may lie in the calculation of the scattering integral using Biersacks MAGIC formula[15] and the ZBL potential. TRIDYN and SDTrimSP use the Kr-C potential, known to be more accurate. Comparison between different modes of computation lends evidence to this, as seen in Fig. 10 in Ref. [21].

Unfortunately, despite these issues, SRIM continues to be widely used. TRIDYN and F-TRIDYN offer more accurate and more fully featured alternatives to SRIM. Work on Python libraries for F-TRIDYN input and output file handling have been developed in order to bring simplify its use.

2.8.2 MARLOWE

MARLOWE is a BCA code that includes a crystalline surface. Thus, MARLOWE is capable of simulating effects such as ion-channeling that amorphous BCA codes cannot[42].

2.8.3 FTRIM and VFTRIM

FTRIM (Fractal-TRIM) and VFTRIM (Vectorized Fractal-TRIM) were developed as upgrades to TRIM. FTRIM was the first BCA code to include a fractal model of surface morphology and is the immediate predecessor of F-TRIDYN[45]. Fractals were included in a manner similar to those discussed in this work. Fractals in FTRIM and VFTRIM were generated using similar fractal generators[45]. VFTRIM, a vectorized version of FTRIM, offered a significant performance enhancement over FTRIM.

2.8.4 TRIDYN

TRIDYN is an upgraded version of TRIM that included a dynamic model of target composition[32]. TRIDYN uses the Kr-C potential to model interatomic forces and forgoes the use of the MAGIC formula for calculating the scattering integral.
2.8.5 SDTRIMSP

SDTrimSP is an upgraded version of TRIDYN that includes features that TRIDYN does not, including surface morphology and lookup tables for atomic parameters. These features have since been implemented in F-TRIDYN, albeit using different methodology.

2.8.6 F-TRIDYN

F-TRIDYN is the code whose development is the primary subject of this work. F-TRIDYN is an upgraded version of TRIDYN that includes explicit fractal morphology, additional options for input and output for file-based code coupling, extensive libraries for generating fractal surfaces and input files, lookup tables for atomic parameters, and surface analysis algorithms. F-TRIDYN reproduces experimental results of the effect of surface roughness on sputtering yields and matches theoretical and semi-empirical models[14].
3 Surface Roughness in Plasma-Material Interactions

3.1 Effect on PMI

Surface roughness plays a significant and as-of-yet not fully understood role in PMI. Surface roughness has a strong effect on sputtering and reflection and may play a crucial role in plasma devices, including future fusion reactors[46]. All natural atomic surfaces have some characteristic roughness[2]. Depending on the nature of a given surface, including its specific morphology and scale, the effect that roughness plays in PMI can change. In this work, the effect of surface roughness on sputtering yields has been investigated. Depending on the energy of the incident ions, the roughness of the surface can either increase or decrease the sputtering yield, especially at high angles of incidence where ions spend more time interacting with the surface as opposed to the bulk. For example, Argon ions impacting a Tungsten surface at 80 degrees will sputter approximately 3.5 times more Tungsten atoms from a rough surface than an equivalent smooth surface.[14]. In a controlled plasma environment, increasing impurity density by such a factor could have a drastic effect on plasma performance. In a fusion environment specifically, because the Bremsstrahlung energy loss term scales quadratically with average plasma atomic number(see Eq. 2.1), any significant increase in sputtering yield from heavy first wall materials could be catastrophic for future fusion reactors as they operate at long timescales.

Fortunately, for the species, energies, and angles relevant to fusion PMI problems, sputtering yields on average have been found in this work to decrease as surface roughness increases. On the other hand, however, the effect of plasma exposure on wall components in a fusion environment, including erosion, can negatively impact material properties, suggesting that rough surfaces may be more prone to emission of so-called dust (large conglomerates of atoms) into the fusion plasma. Understanding the interplay between these two effects requires coupling of a BCA code, which operates on atomic timescales, and other codes that can handle long timescales and macroscopic material effects.
3.2 Computational Surface Models

3.2.1 ”Ripples” Model

One model of surface roughness is to create a rough surface composed of circular ripples[50]. This model approximates the atomic surface as a series of semicircular ripples connected to each other using smooth curves. This model is predictive of two properties seen in sputtering yield results from F-TRIDYN and other surface roughness models, namely, a decrease in sputtering yields at low angles of incidence and a shift to the right of the angle of maximum sputtering yield[14]. These properties have also been observed experimentally, suggesting that they are a common property of the sputtering yields of rough surfaces[26].

This model is attractive for use especially in theoretical and analytical models of sputtering, as it can be incorporated into classical sputtering theory. Including this model of surface roughness into Sigmund’s theory allows for the calculation of normalized sputtering yields for rough surfaces. A weakness of this model is that its free parameters are not directly measurable. This means that, in order to calculate sputtering yields, the free parameter must be fit to experimental data. The predictive power of this model is thus not as strong as a model with a directly measurable free parameter controlling the roughness of the atomic surface.

3.2.2 Local Angle-of-Incidence Distribution Model

A local angle-of-incidence distribution model seeks to approximate a rough surface by reducing an explicit surface to a distribution of local angles of incidence incoming ions experience upon impact. This model is semi-empirical and relies on the ability to image rough surfaces in detail in order to reconstruct them and produce the local angle-of-incidence distributions depending on the angle of the incident ions[26]. This method reproduces previously reported effects on sputtering yield dependence on angle of incidence, namely the decrease in yield and a shift in the peak of maximum sputtering. However, this method is not capable of true predictive power, because it requires analysis of a physical surface before it can be used. It is possible to construct local angle-of-incidence distributions for any rough surface model, and this may allow for a connection between this model of surface roughness and others directly through these distributions.
3.2.3 Tessellated Model

Another surface model, developed for use in SDTrimSP, uses a tessellated surface composed of trapezoidal hills and valleys whose scale is controlled by a single free parameter\[8\]. This surface model seeks to recreate regular, repeating atomic structures that are a result of the underlying lattice structure of the target.

3.2.4 Fractal Surface Model

Fractals in nature are ubiquitous. Atomic surfaces can be well represented by fractal surfaces in BCA codes. As a model of surface roughness, fractals are attractive for their useful properties, such as straightforward creation of arbitrarily complex surfaces via the fractal generator method, ease of measurement of fractal dimension both experimentally and computationally, and their inherent similarity to natural surfaces. Fractal models of surface roughness were introduced in BCA codes with the FTRIM code\[45\]. The fractal model of surface roughness is used in this work as the primary surface roughness model in F-TRIDYN. This model uses explicitly generated fractal surfaces and uses the fractal dimension of the surface as a free parameter. This model is attractive because the fractal dimension is a measurable parameter of a surface. Fractal theory and algorithms for inclusion in computer simulations are discussed in Chapter 4 of this work.

3.3 Modeling Surface Evolution

Surface evolution in ion-material interactions is often driven directly by atomic motion post-impact of the incident ion. This variety of surface evolution was modeled in the BCA code VFTRIM\[47\]. In VFTRIM, Primary Knock-on Adatoms (PKAs), the atoms set into motion directly by incident ions, are tracked throughout their simulations and their final positions used to measure a fractal dimension representative of the resulting surface post-bombardment via binning of average heights of the final atomic positions. This new surface could then be used as input to the next simulation, allowing surfaces to evolve over fluence steps. If there are no thermal effects, this method allows for modeling the evolution of surfaces under ion-irradiation and plasma-facing conditions.

In F-TRIDYN, positions of both PKAs and Secondary Knock-on Adatoms (SKAs), or the atoms that are set into motion by collision with PKAs, are tracked. Their final positions are output in a similar way to VFTRIM. In
F-TRIDYN, however, surface reconstruction is not handled via binning but rather via a custom stepping algorithm that simultaneously reconstructs and measures the fractal dimension of the resulting surface. This algorithm is described in Chapter 4. Concisely, this code uses a stepping algorithm to walk across the noisy set of PKA positions representing the new surface with a defined length step. Fig. 4.10 shows a demonstration of the algorithm on a simulated set of PKA positions based on a fractal curve with a known fractal dimension. F-TRIDYN allows for the use of either the reconstructed surface directly or a fractal analogue created therefrom as input for its surface model.
4 Fractals

4.1 Fractal Theory

Fractals are complex and fascinating mathematical objects that are ubiquitous in nature. Phenomena as varied in field and scope as atomic surfaces[3], flow through porous media[16], DLA clusters[4] and coastlines[30] can be represented using fractals. No precise mathematical definition of a fractal exists[1], but a thorough understanding of their properties in nature and mathematics has been developed in the decades since Mandelbrot coined the term and collected the state of the art research in the field in his 1982 book The Fractal Geometry of Nature[31]. Mandelbrot’s revelation was to connect the ideas of self-similar sets in mathematics to self-similar objects in nature.

All fractals, natural and mathematical, share certain unique properties. First, they are self-similar. Self-similarity implies that subsets of an object resemble the whole of the set, and vice versa. A simple, everyday example of self-similarity can be found in nature in the form of plants such as trees. A branch cut from a large tree will have much the same structure as the entire tree. It will have a large, main branch, analogous to the trunk, from which smaller branches emerge in a pattern similar to, albeit at a smaller scale, large limbs emerging from the trunk of the tree. Other everyday examples of this sort of self-similarity can be found in rivers and their tributaries, coastlines, and bolts of lightning. Figure 4.1 shows some examples of natural fractals. Fig. 4.1(a) shows a complex atomic surface formed via plasma-material interaction. Specifically, this surface is the result of incident energetic Helium ions diffusing and reacting with a tungsten lattice. Over time, this interaction leads to the structure known as tungsten fuzz. Tungsten Fuzz has been measured to have a fractal dimension[23]. Fig. 4.1(b) shows a plant in the Brassicaceae family, commonly known as Romanesco broccoli, which has been bred to form a fractal-like flower which is valued for its taste and aesthetic properties. Fig. 4.1(c) shows another atomic surface, graphitized Carbon, which has a measurable fractal dimension[3], measured, and studied. Simple mathematical fractals are commonly used as teaching aids.
for difficult concepts such as measure theory and chaos. One such example is the Cantor set, a subset of the real numbers on the interval $[0,1]$. The Cantor set, first discussed by its eponym, Georg Cantor, the mathematician who invented set theory, is a simple example of a mathematical fractal.

The Cantor set was of interest to Georg Cantor for its unique properties. Specifically, it is an uncountable set. This means that the size of the set is the same as that of all the real numbers. Second, it has measure 0. This means, in layman’s terms, that despite containing uncountably many members, the set has zero length. Simultaneously the Cantor Set seems to share properties of 0-dimensional objects (i.e., points) and 1-dimensional objects (i.e., lines). Therefore, the set seems to simultaneously behave like 0-dimensional and 1-dimensional objects. At first, this seems an impossibility, but the apparent paradox is resolved with the introduction of a measurement of dimension between zero and one. This is identifiable as a fractional dimension, a term from which the word fractal was derived. A fractional, or fractal, dimension can have non-integer values, and represent objects whose complexity places them somewhere in between objects with integral dimension. An object with a fractal dimension between 0 and 1 therefore has the same measure as a 0-dimensional object yet more complexity. An object with a fractal dimension between 1 and 2 will have the same measure as a 1-dimensional object, yet exhibit more complexity than a line. Its total length will depend on the magnification at which it is measured. All true fractals have a fractal dimension, as a non-integer dimension is in effect a measurement of how complex a set is upon magnification. The integer part of the fractal dimension refers to the dimensions wherein a given fractal is embedded, and along which its size can be measured. The fractional component therefore measures the further complexity beyond the so-called embedding dimension that the fractal exhibits. A Cantor set can be simply constructed via an infinite series of recursive iterations, one must iteratively remove the center third of each piece of the current set. Successive iterations of this process are referred to as $K_n$. The first iteration, $K_0$, is simply the real number line between 0 and 1. $K_1$ is the set of numbers $[0, 1/3] \cup [2/3, 1]$. $K_2$ is the remainder of $K_1$ after the middle third is removed from each of the two intervals in $K_1$. This process, completed ad infinitum, will produce the Cantor set. Specifically, this formulation of the Cantor set is known as the Middle-Third Cantor set.

Figure 4.2 shows the first 7 iterations of the Middle-Third Cantor Set. Continuing this process infinitely results in a true fractal. Stopping the iterations at some finite number will produce an object that has fractal
properties, but they will be limited to a specific range of magnification, similarly to fractals in nature.

4.2 Fractal Dimension

Fractal dimension can be thought of as a measure of the complexity of a set beyond the dimension in which it is embedded, as discussed in the previous section. Specifically, a fractal dimension can be thought of as a measure of a scaling law that predicts how a set is measured at different levels of magnification. That scaling law can be expressed in Eq. 4.1, where \( D \) is the fractal dimension, \( C \) is a proportionality constant, and \( a \) is the scale of magnification.

\[
L = Ca^D
\]  

(4.1)

Multiple fractal dimensions have been defined and used, but the simplest and most often used is the box counting dimension, also known as the box dimension. Other fractal dimensions include the Kolmogorov dimension and the Minkowski dimension, but in almost all applications the three have equivalent value for any given fractal[43]. The box dimension is the simplest measure of how a set or object scales at differing levels of magnification, and is therefore the simplest measure of fractal dimension. This is achieved by defining a grid of usually (but not necessarily) uniform boxes, of side length \( \epsilon \), and counting continuously how many of the boxes the set occupies as the side length goes to zero. Mathematically, the box counting dimension of a bounded set \( S \) in \( R^d \) is defined in Eq. 4.2 when the limit exists[1].

\[
boxdim(S) = \lim_{\epsilon \to 0} \frac{\log N(\epsilon)}{\log 1/\epsilon}
\]  

(4.2)

For ease of calculation, it is sufficient to use a discrete series to represent the side length \( \epsilon \), denoted \( b_n \) as long as the conditions shown in Eqs. 4.3 & 4.4 is met.

\[
\lim_{n \to \inf} b_n = 0
\]  

(4.3)

\[
\lim_{n \to \inf} \frac{\log b_{n+1}}{\log b_n}
\]  

(4.4)

Depending on the object whose dimension is being measured, this replacement of \( \epsilon \) with a discrete series whose limit can be easily calculated can significantly simplify the theoretical calculation of the fractal dimen-
sion. This allows one to calculate the fractal dimension directly of many mathematical fractal objects, such as the Cantor set. Measuring the fractal dimension of the Cantor set reduces to a problem of choosing the correct series for $\epsilon$ in $n$, a simpler problem than finding a function for $\epsilon$ that depends on the magnification. As an example, the fractal dimension of the Middle-Third Cantor Set is easily measured\cite{1}. Each iteration of the Cantor set, $K_n$, is made up of $2^n$ intervals. Each of these intervals is of length $1/3$. Using boxes of side length $\epsilon = 1/3$, it is obvious upon visual inspection (see Fig. 4.2 that 2 of 3 boxes will be filled each iteration. Using this knowledge to define the magnification $\epsilon$ as $1/3^n$ yields the correct fractal dimension of $\log 2/\log 3 = 0.631\ldots$

To calculate the box-counting dimension, boxes may be any appropriate dimension shape, e.g. circles, regular polygons, etc. For example, circles have been used to measure the coastlines for fractal dimension\cite{30}. For ease of determining computationally whether a point or line lies within a box, square boxes are used throughout F-TRIDYN and its associated works without loss of generality in its measurements of fractal dimension. This allows the determination of whether a particular box contains an element of the set to be measured to be simplified to only as many simple, Boolean loops over spatial dimensions as the embedding dimension.

Box dimensions of physical and simulated fractals can easily be approximated by counting the number of boxes occupied at several levels of magnification and finding the slope of that line on a log-log plot with respect to the change in the inverse side length. This can be seen by examining Eq. 4.1, the fractal scaling law. On a log-log plot, the slope of the line will be equal to $D$, previously identified as the fractal dimension. That this dimension and the box-counting dimension are equivalent is beyond the scope of this work. It is sufficient to say that using this method has proven accurate to any desired degree of accuracy in all applications investigated herein, including the example in Fig. 4.3. That figure shows the previously described fractal scaling law for a rendered Koch Snowflake, a relatively famous fractal for its pleasing aesthetic properties. A rendered Koch snowflake and its true, theoretical fractal dimension are contained in the inset box. The line and its slope resulting from the box-counting algorithm are shown, demonstrating the accuracy of the approximate method for even low-resolution rendered fractals. It is possible to gain as much precision as one desires by rendering the fractal at higher resolution and using smaller and smaller boxes in the algorithm. In this example, the snowflakes size was restricted to $[0, 1] \times [0, 1]$ in the $R^2$ plane.
Implementation of this approximation to the box counting algorithm on a computer is straightforward, but when used to measure natural phenomenon there are potential pitfalls. Many natural phenomenon are fractal in nature, but only over a limited range of length scales[43]. Non-fractal behavior can be induced if the log-log plot of filled boxes versus inverse side length is non-linear. For example, the cross section of an atomically rough surface may have a box dimension of $D = 1.3$ for side lengths on the scale of hundreds of angstroms, the natural unit of atomic lengths. However, that same surface may have a fractal dimension of $D = 1.0$ (the same as a line) for side lengths on the scale of centimeters. Another example of nonlinear behavior in the log-log box counting plot is exhibited by shapes such as circles. A circle at no range of length scales has a linear log-log box-counting plot, as shown in Fig. 4.4. Although over a limited range of side lengths the measured box counting dimension may be non-zero, examination of the entire relevant range of side-lengths is necessary to observe that it is not a fractal. However, over a range of relevant length scales, the coastlines and land-boundaries of the countries included in the figure show that they are true fractals, over a range of length scales.

An important property of the box dimension is that the box dimension of the cross of two sets, $A$ and $B$, each in $\mathbb{R}^n$, is that the sum of the box dimension of each set in $\mathbb{R}^n$ will be a new set, $AXB$, with a box dimension between $n$ and $n + 1$, as shown in Eq. 4.5.

$$boxdim(AXB) = boxdim(A) + boxdim(B)$$ (4.5)

This property is used in F-TRIDYN to reduce the dimensionality of the surface from 3 to 2 dimensions without loss of generality. This can be done by assuming that the box dimension of the computational surface is 1 in one direction and $D - 1$ in the other, where $D$ is the fractal dimension of the 2-dimensional surface. Via this rule, the box dimension of the cross of a flat surface with the surface used in F-TRIDYN will be $D$, the fractal dimension of the real surface minus 1. As long as the incident ions have an angular distribution restricted to the plane, this assumption is valid, as each incident ion will interact with an appropriately rough surface with the correct fractal dimension. This method was also used in the previous codes VFTRIM and FTRIM, along with other methods[44][45][46]. If the incoming distribution of ions is non-planar, however, this method will fail. Alternate methods include using a statistical model in the other surface dimension (z in the case of F-TRIDYN) and using a second set of fractal surface points to pick
an average surface height.

4.3 Measurement of Fractal Dimension

Since fractals are ubiquitous in nature, the development of methods to measure fractal dimensions of real objects has historically been an important problem. Visual methods are the most straightforward. Imaging techniques exist for nearly all observable length scales, from electron microscopy for atomic-scale fractals such as rough surfaces to satellite imagery for continent-sized fractals such as natural coastlines. At the atomic scale, electron microscopy, ion microscopy, and atomic force microscopy dominate the field of imaging techniques. After a physical fractal has been imaged using one of these techniques or an alternative method, a computer can be used to measure the fractal dimension of the object using the box counting algorithm, Eq. 4.2 and the fractal scaling law, Eq. 4.1. In terms of PMI, the area of use for BCA codes such as Fractal TRIDYN, imaging techniques have significant downsides. Most experiments in this field, especially in the study of fusion-relevant PMI, do not have built in capabilities for atomic-scale imaging. In order to image a surface modified via plasma interaction, it must be brought from the machine to a facility where it can be imaged. This process is referred to as breaking vacuum and is likely to result in some exposure to atmospheric pressure. Exposure to atmosphere can significantly damage or alter fragile nanostructures modified via PMI, and thus visual imaging techniques are not ideal for the measurement of atomic-scale fractal surfaces except in experiments with in-situ diagnostics including imaging.

An attractive alternative to imaging methods for measuring surface roughness is gas adsorption methods. Two main theories of gas adsorption, Brunauer-Emmett-Teller (BET) and Frenkel-Halsey-Hill (FHH) theories have been extended to include a fractal description and thus may be used to calculate fractal surface roughness from gas adsorption experiments[38]. Both theories rely on the observation that over the range of length scales where a surface is fractal, the number of molecules adsorbed onto the surface needed to cover it completely scales with $D$ as shown in Eq. 4.6[3], where $N_m$ is the areal density of adsorbed molecules, $a$ is the size of the molecule used, and $C$ is the surface area of the material. This is directly analogous to Eq. 4.1, the fractal scaling law.

$$N_m = Ca^{-D}$$

(4.6)
In this physical situation, the fractal dimension is being measured by using different-sized adsorbing molecules as the boxes to cover the surface and thus measure a fractal dimension. This method succeeds because, as previously discussed, boxes used in the box-counting algorithm need not be square. Thus, the molecules used may be of any shape. This form of the fractal scaling relation is convenient for inclusion in the existing BET and FHH theories of adsorption. These two formulations allow one to back-calculate the fractal dimension of a surface if the other experimental parameters are known. By using adsorbing films with different sizes, an average fractal dimension of the substrate can be found through the fractal scaling law, equation, Eq. 4.1. This allows one to make live measurements of fractal dimension in-situ, without breaking vacuum. Most PMI experiments include appropriate apparatuses for adsorption experiments, requiring only gas inlet and an RGA or other gas analysis tool. These properties make gas adsorption methods very appropriate for measuring fractal dimension for atomic-scale experiments.

4.4 Fractal Generator Algorithms

The fractal generator method was first described by Mandelbrot as a technique to create curves with fractal properties[30]. Fractal generators are curves made of piecewise straight line segments. Fractals can be created from fractal generators using an iterative technique. First, for each line segment in the initial generator, a copy of the original generator is uniformly scaled and rotated via rotational matrices so that it spans that line segment. Then, each line segment is replaced with the appropriately scaled and rotated copy of the original fractal generator. On the next iteration, the same procedure is followed for each of the resulting new line segments. The number of line segments increases as $N^2$, where $N$ is the number of line segments in the original generator. This process allows one to create self-similar, complex mathematical objects in any dimension with nearly any fractal dimension desired at any length scale and with arbitrary structure. As such, they represent the most powerful tool in the understanding and use of fractals in computational simulations.

To simplify calculations, fractal generators researched for applications in computer simulations can be created with the following properties:

- Original generator is of length 1 in x
- Composed of line segments of equal length
• Deviations from the horizontal restricted to equilangular line segments at an angle $\beta$

These properties were adhered to for all fractal generators investigated for use by F-TRIDYN. Each of the generators presented in Mandelbrot’s paper meet these properties$^{[30]}$ and his figure showing his generators has been included as Fig. 4.5

The first requirement was chosen because it simplifies the calculation by removing terms for the length of the generator in x from the construction algorithm at all steps. The second and third requirements were chosen because they allow for the simplification of the algorithm by pre-calculation of cosines and sines of the angle by which all non-horizontal line segments deviate, $\beta$. The third requirement also allows for the simplification of the encoding of the generator shape, an original technique developed for this thesis work. Generators can be fully recreated simply by use of a signed binary array with as many elements as line segments in the generators. If an element is 0, the line segment is horizontal. If an element is 1, then the corresponding line segment deviates from the end point of the previous segment by the angle beta. If an element is -1, then the line segment deviates by an angle of negative beta. Using this array, the algorithm can be reduced to an array-wise multiplication for each line segment at each iteration. This algorithm is capable of creating any fractal curve with a generator that meets these requirements. Fig. 4.6 shows several iterations of a fractal generator, along with the signed binary array that represents it, in detail.

Fractals resulting from this process can subsequently be scaled to any corresponding length scale desired. If these simplifications are made, the box dimension of a generated fractal using this method can be calculated via Eq. 4.7, where a is the number of straight, horizontal line segments of the original generator and b is the number of line segments that deviate from the horizontal at the angle $\beta$. This formula can be derived from Eq. 4.2 using the series representation of epsilon in Eq. 4.8 and the recognition of Eq. 4.9 as the number of occupied boxes of side length $\epsilon$ for all $n$.

$$D = \frac{\log a + b}{\log a + b \cos \beta} \quad (4.7)$$

$$\epsilon = \frac{1}{(a + b \cos \beta)^n} \quad (4.8)$$

$$n = (a + b)^n \quad (4.9)$$
This derivation shows some of the power of the fractal generator method. Taking Eq. 4.7, one can solve for $\beta$ and determine, for a given generator encoded using a signed binary array, the angle necessary to produce any fractal dimension with that fractal dimension (up to $1 + D_g$, or 1 plus the integral dimension of the original generator). One downside of this technique is that it is possible to construct fractals that are not practical to use. In particular, it is possible to construct fractal curves that have points that lie outside of the bounded interval $[0, 1]$. It is also possible to construct fractal curves that include self-intersections. These pitfalls can be avoided with careful choice of the correct fractal generator for a given problem. Avoiding these pitfalls, generators provide F-TRIDYN with a fast, practical method for the generation of explicit fractal curves.

### 4.5 Fractal Generators

Fractal generators are used in F-TRIDYN for 3 reasons: first, they provide a fast and consistent way to produce a fractal with arbitrary scale and fractal dimension. Second, they have useful geometric and statistical properties. Third, they can be made to be volume conserving, an important property for simulation purposes. The generator used specifically will henceforth be referred to as the bounded trapezoidal generator, as seen in Fig. 4.6. It has the following properties: It is composed of 9 equal-length straight line segments. If the segments are numbered 1-9 from left to right, odd numbered segments are horizontal. Even numbered segments are, to their sign, equiangular with characteristic angle beta with respect to the horizontal. Segments 1, 5, and 9 remain stationary for all angles beta. Segment 3 rises above the horizontal and segment 7 sinks below the horizontal. This generator is rotationally symmetric and conserves the area below it. As described in the previous section, the signed binary array corresponding to this fractal generator is $[0, 1, 0, -1, 0, -1, 0, 1, 0]$. It is immediately obvious from the encoding array that this generator is volume-conserving, as it is rotationally symmetric about its center. This means that as the fractal dimension is increased, an equivalent volume is gained to the left of the center segment (which remains horizontal) as is removed to the right of the center segment. This keeps the mean height of the surface at zero, an important property chosen to keep the particle initialization conditions from TRIDYN the same.

Other volume conservative fractal generators were investigated. First, the unbounded trapezoidal generator, shown in Fig. 4.7(b) Encoding this generator as a signed binary array produces: $[1, 0, -1, 0, -1, 0, 1, 0]$. It is clear...
form this encoding that this generator is identical to the bounded trapezoidal generator with the first and last line segments removed. This generator was not used because it has points that, as the fractal dimension is increased, spread left past 0.0 and right past 1.0. Although this does not have an effect on the correctness of the mathematical representation of the surface in F-TRIDYN, these points will not be used as they are outside the boundaries for the simulation. Additionally, the unbounded trapezoidal generator can be extended to fractal dimensions up to only 1.5 (to one significant figure), without self-intersections. Although F-TRIDYNs surface will remain correct even for surfaces with self-intersections, such surfaces no longer describe any physically occurring surface and are thus avoided.

Second, the bounded triangular generator, in Fig. 4.7(a), was not used because it does not share qualitative features with real rough surfaces. Its encoding in the signed binary array is [0, 1, -1, 0, -1, 1, 0]. Additionally, the bounded triangular generator has a smaller RMS proportionality constant, as discussed in the next section. Essentially, this means that the bounded triangular generator creates surfaces with little overall variation in height across its length and is unrepresentative of real rough atomic surfaces. For these reasons, all results of F-TRIDYN have been produced using the bounded trapezoidal generator for consistency. Other generators produce similar results for similar fractal length scales and corresponding fractal dimensions.

4.6 Fractal Surface Algorithms

Fractal surfaces in F-TRIDYN are input as a series of ordered, piecewise, and real-valued y-z coordinates. Surfaces are constructed using the algorithm described in the previous section. The main role of the surface is determining if coordinate pairs (representing positions of collision partners) are inside or outside of the surface. If the collision pair is outside the surface, the impact parameter is set to an arbitrarily high value, such that the interaction effectively does not occur. If the collision pair is determined to be inside of the surface, however, the impact parameter is chosen according to the material model. In the case of the amorphous material in F-TRIDYN, the impact parameter is chosen from a square root distribution between 0 and the maximum impact parameter $p_{\text{max}}[15]$, it will be handled correctly by F-TRIDYN. Complex surfaces with voids and overhangs are of increasing interest in many PMI fields, including materials engineering and fusion engineering. These surfaces may be desired, for which the dynamics of their
genesis must be understood, or detrimental to device performance, for which their dynamics must be understood to prevent their occurrence. Determination of whether a coordinate position is inside or outside of the surface is done using an algorithm based on the Jordan Curve Theorem. The Jordan Curve Theorem states that every closed, non-self-intersecting curve divides the space in which it is embedded into two regions, an exterior and interior region. The determination of whether a given point in the embedding space lies in the exterior or interior region of the curve is not a simple problem. However, a corollary of the Jordan Curve Theorem provides a method to determine whether points are inside or outside of a shape. Given a shape made of straight or curved lines and a point (which is definitionally a Jordan Curve), if a ray is drawn from the point in any direction to infinity, it will intersect with the shape N times, where N is a finite natural number. This corollary to the Jordan Curve Theorem states that if N is even, the point is outside of the polygon. If N is odd, the point is inside the shape. In computer graphics, problems of this nature are referred to point-in-polygon problems, referring specifically to the case where the Jordan Curve in question is constructed from straight-line segments. These curves are the simplest to utilize computationally, and are thus used in F-TRIDYN. Their use in computer graphics provided an use-case for highly efficient solutions to the PIP problem, inspiring this work.

Since the ray may be passed in any direction, it is convenient to pass it upwards in x, out of the surface. This way, the sides and bottom of the polygon, representing the boundaries of the surface in the code, do not need to be included. An algorithm using this method to determine whether points are inside or outside of a polygon has been implemented in F-TRIDYN and will henceforth be refereed to as the even-odd algorithm. An example of this algorithm, along with counted intersections, is shown against a F-TRIDYN fractal surface to demonstrate how the ray is cast and intersections are counted in Fig. 4.8.

Before using the even-odd algorithm, a simple bounding box check can be performed. If the particle is above the highest point of the surface, it is outside of the surface. When using surfaces with great variation in height for systems with significant sputtering, this saves computational time by skipping over the even-odd algorithm for particles that have left the surface but have not yet reached the sputtered height threshold. In F-TRIDYN, the following code is used to count the number of intersections of a ray cast from a coordinate triple in F-TRIDYN:

\[
\texttt{fractal}
\begin{align*}
\texttt{loop: do } & i=1, \text{nfracPoints} - 1
\end{align*}
\]
if ((fracY(i).gt.y).neqv.(fracY(i+1).gt.y)).and. &
(x.gt.(fracX(i+1)-fracX(i)) * (y-fracY(i)))/ &
(fracY(i+1)-fracY(i))+fracX(i))) then
  inside2=inside2+1
endif
end do fractal_loop

The Boolean statement in the loop over coordinate positions of particles
can be broken into the following individual statements.

y_fractal ,i>y_particle != y_fractal ,i+1>y

This statement checks if the particle is in between the left and right
bounds of the ith line segment of the fractal surface.

x > (x_fractal ,i+1-x_fractal )*(y-y_fractal ,i)/ &
(y_fractal ,i+1-y_fractal ,i)+x_fractal

This statement checks that the line y = 0, representing the ray cast
upwards, crosses the ith line segment of the fractal surface. The even-odd
algorithm is of complexity \( O(N) \), where \( N \) is the number of points in the
surface. This is much improved performance compared to Fractal TRIM
and VFTRIM, which each had an algorithm with worst-case performance
of \( O(N^2) \)[46]. This translates to a significant real-world computational performance enhancement. Additionally, this algorithm is easily parallelized,
since each line segment can be counted independently and then each threads
number of collisions can be summed at the end of the loop to reach the de-
sired result. This has been done using OpenMP* compiler directives, and
improves whole-code performance by 15-20 percent. Fig. 4.9 shows a demonstra-
tion of the accuracy of this algorithm in determining whether a given
coordinate point is inside or outside of the surface. Points that lie directly
on the line are treated consistently as lying outside the surface. This demonstra-
tion used 10,000 points and took less than 1 minute to compute on a
laptop computer, highlighting the efficiency of the even-odd algorithm.

In addition to the fractal-type surfaces used in F-TRIDYN, the algo-
rithm can accept any explicit piecewise surface. This allows for the use of
the BCA to simulate surfaces analyzed from real systems as opposed to the
generated surfaces presented here. Additionally, structured surfaces such as
trenches and nanostructures at the atomic scale will have an effect on sput-
tering yields and the angular and energy distributions of sputtered particles.
Surface modification techniques to intentionally change these qualities may
be important for plasma processing device engineering or first wall and di-
verter design for the future.
Surface evolution has been studied using the fractal-BCA model has been achieved in the codes FTRIM and VFTRIM\cite{47}. Conceptually, FTRIDYN is similar to these codes in that it is a BCA ion-irradiation code that includes an explicit fractal surface. Surface evolution in cold systems where the primary driver of atom mobility is ion-atom collisions can be modeled using the fractal-BCA model. Modeling surface evolution in FTRIDYN requires several discrete additions to the code base. First, Primary Knock-on Adatoms (PKAs), the atoms displaced directly by the incident ions, must be tracked in space, as their positions will largely determine the final state of the surface. Second, a method for analyzing the resulting PKA data is needed. Third, a means of sending a new surface to the code every surface-evolution time-step is also needed.

Tracking PKAs in F-TRIDYN is handled at the end of the particle output cycle. If a particle has been identified as one displaced by an incident ion, it is assigned a flag that identifies it is a PKA. At the end of the simulation, after all PKAs and other atoms in motion have reached the cutoff energy (EF in the TRIDYN user manual), the spatial location of each tracked PKA in x, y, and z is output to disk in the form of a space-delimited text file that includes identifying information for each particle, including the atomic number and internal F-TRIDYN reference number for that species.

To analyze the resulting noisy surface post-simulation, additional code is required. This code simultaneously reconstructs and measures the fractal dimension of a noisy surface such as those created by tracking of PKAs in F-TRIDYN. The code requires the following assumptions:

- The surface is represented by a curve in the y-x plane that begins at the origin and ends at the same location in the x-axis
- The surface can be represented by a series of piecewise line segments
- The surface does not self-intersect or travel backwards upon itself from left to right

With these assumptions, the code can at once measure the fractal dimension and provide reconstructions of the resulting surface that the PKAs represent. This is accomplished using an iterative stepping algorithm. The iterative stepping algorithm reconstructs the surface for various ruler lengths. This process is analogous to the box-counting method of measuring fractal dimension, in that the total length of the resulting curve can be plotted against the ruler length on a log-log plot to determine the fractal dimension. As the ruler lengths used in this process decreases, the reconstruction of the
Surface becomes more accurate until it reaches the interatomic distance, at which decreasing the ruler length no longer produces surfaces representative of atomic structures.

Reconstructing the surface for each ruler length involves a complex weighting system. First, the initial point is chosen. Using the assumptions listed before, the initial point is always the origin. At the first step, the code determines the angle at which the next line segment is most likely to represent the actual atomic surface. This is done via 2-dimensional weighted counting of nearby PKAs. First, counting is weighted via an angular distribution of the form in Eq. 4.10

$$f_\theta = \frac{(\cos (\theta - \theta_{\text{max}}) + 1)^n}{2^n}$$ (4.10)

This weighting prevents the reconstructed curve from traveling backwards upon itself in regions where final PKA positions are dense. An $n^{th}$-order cosine distribution is a natural choice for restricting steps to be forward facing, since it has zeros at both the maximum and minimum angles of deviation from the previous direction. The number of PKA positions along this angle is counted and multiplied by the $n^{th}$-order distribution. In this distribution, $n$ is a free parameter that can be used to fine-tune the behavior of the weighting. For systems and energies relevant to fusion materials, $n = 2$ was found to most accurately reproduce fractal dimension measurement and was thus chosen to reconstruct surfaces in F-TRIDYN.

Second, counting is weighted in $r$ via an $\exp(-x^n)$ distribution, as shown in Eq. 4.11

$$f_r = \exp(-\frac{d^n}{r^n})$$ (4.11)

$\exp(x^n/r^n)$ distributions are a natural choice ($n = 2$, for example, is a Gaussian distribution) for weighting particle counting in the context of natural noise. Weighting in $r$ prevents artifactual structures caused by restricting counting to a threshold around the next stepping point. For F-TRIDYN, $n = 2$ was chosen for this distribution because without any other evidence the normal distribution is a proper choice for accounting for noise in particle positions. Using these two weighting distributions, particles are counted for each next possible step location. As the algorithm steps along the PKA distribution, it constructs a piecewise curve that is a likely reconstruction of the actual surface that the final PKA positions represent. This curve is used, along with the convenient measurement of fractal dimension that is provided by decreasing the ruler length for stepping reconstruction
and using the box-counting algorithm, to provide the next surface for F-TRIDYN to process. A reconstruction of an artificially noisy surface can be seen in Fig. 4.10. Using this methodology, resulting surfaces at each surface evolution time-step can be fed to F-TRIDYN to produce results of evolving surfaces in the fractal-BCA framework.

4.7 Statistical Properties of Generated Fractals

One interesting property of fractals generated using the method outlined above is that the distribution of area below the fractal curve, representing a volume of material, along the y-axis has well-defined statistical behavior. By definition, the mean of the distribution of the points that make up the surface is zero. The distribution of points is non-normal, and a histogram of this distribution, taken from a 4th-iteration fractal created using the bounded trapezoidal generator is shown in Fig. 4.11. The variance of the points that make up the surface scales as the square root of the fractal dimension. Similar scaling is found in the variance of Brownian motion with respect to time.

This opens up the possibility of approximating the complex fractal surface with what amounts to a probability mass function. Such a function would have the following properties:

- It would be equal to the normalized integral of the area under the curve to a depth $x$.
- It would be zero at zero, representing a zero probability of a point above the tallest part of the surface being inside the surface.
- It would be 1.0 at infinity, representing a certainty of a point being inside the surface below a certain depth.
- It would have a variance that depends on the fractal dimension and the length scale of the fractal generator it is approximating.

It is possible to construct exact probability mass functions for fractal generator curves. However, it is very difficult. In the first generation, the PMF resembles a piecewise staircase function, which is simple to reconstruct for any given generator and dimension, as seen in Fig. 4.12(a). After the second generation, however, the PMF exhibits fractal-generator-like behavior, as seen in Fig. 4.12(b). The whole PMF curve of the first generation is replicated onto each section of the original PMF. While this resembles
the method for constructing the fractal generator itself, because the PMF is representative of the depth integral of the surface, the recursive algorithm does not capture the complicated behavior accurately. Although the second-generation fractal PMF has been found numerically using the genetic algorithm curve fitting software Eureqa, this process is complicated and restricted to the replication of a fractal PMF for one specific generator at a specific angle only. To find the numerical PMF, dependent on the fractal dimension and the fractal length scale, the PMF was integrated numerically from a rastered fractal curve. The numerical PMF was calculated for 100 different fractal dimensions spaced evenly between 1.0 and 1.8 (the range, to one significant figure, of fractal dimensions for the bounded trapezoidal generator that do not lead to self-intersections). Each point of the second-generation PMF was tracked as it moved in the depth-dimension axes with increasing fractal dimension. These curves were then fit to power-law polynomials using the aforementioned genetic algorithm curve-fitting software Eureqa. This method was not used for F-TRIDYN, as this produced PMFs only for first generation fractal surfaces. These surfaces have flat planes much larger than the mean interatomic distance and do not approximate physical surfaces.

This method has significant weaknesses. Because it is limited for practical reasons to the first-generation PMF (as the number of points in the PMF increases exponentially with each generation), it does not capture the relative smoothness of the full fractal PMF as seen in Fig. 4.12(c). Additionally, this method is not practically expandable to other generators, limiting its usefulness. Finally, this method does not produce a functional form of the PMF, but rather a piecewise curve. Piecewise curves can more computationally expensive to implement than smooth functions, depending on the compiler and function in question.

Another approach is to model the PMF using an error function. This is equivalent to approximating the distribution of points of the surface with a normal distribution. This approximation has been found to be capable of reproducing results for sputtering yield of the explicit fractal model, as discussed later in this work. Additionally, this models roughness parameter, RMS roughness, can also be directly measured from the surface using electron microscopy or other atomic-scale diagnostic techniques. In this way, RMS roughness serves as an alternate single parameter to describe the roughness of the surface that the ions interact with. The use of the explicit surface, fractal or not, and the statistical model of the surface, will depend on the application. Highly structured surfaces may be better represented by
an explicitly patterned surface, whereas surfaces with approximately random roughness may be better represented using a statistical model. Either way, this work provides the tools to link the concepts of fractal dimension and RMS roughness as they relate to 2-dimensional surfaces as those used in F-TRIDYN. For F-TRIDYN this provides a methodology for comparing previous experiments, which do not use fractal dimension as a measurement of surface roughness, to the explicit fractal surface model.

### 4.8 Link to RMS Surface Roughness

An important link to physical surfaces is the RMS roughness. RMS roughness is the root-mean-square measure of the variation in height of a surface. Measurements of RMS roughness will vary depending on the length scale of the measurement. Definitionally, RMS is a one-dimensional measure of surface roughness, and as such it is not capable of explicitly capturing complex surfaces with gaps and overhangs as seen on the atomic scale. However, in a Monte Carlo simulation, using the RMS roughness as the standard deviation of the surface as represented using a distribution function is still capable of reproducing the physics of rough surfaces on the atomics scale. The RMS roughness is defined as follows in Eq. 4.8.

\[
R_{RMS} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} y_i^2} \tag{4.12}
\]

Where \(R_{RMS}\) is the RMS measure of surface roughness and \(y_i\) is a set of \(n\) surface height measurements. Fig. 4.13

Given a measurement length scale, the RMS roughness is relatively simple to measure ex-situ. For this reason it is important to connect the fractal picture to the statistical one. Many past investigations use RMS roughness as a standard roughness parameter, and connecting the two frameworks allows for the comparison of the fractal model to past experiments and simulations that do not include a measurement of the fractal dimension. Additionally, the standard deviation of the heights of a given fractal is among the most important for physical processes such as sputtering at normal incidence.

The standard deviations of fractal generators were calculated numerically, by rendering and rastering a 2-dimensional image of the fractal surface and counting pixels. This method is also used to determine the probability of a point at some depth \(x\) being inside the surface given a fractal.
In Fig. 4.14, the RMS roughness of the three fractal generators shown in Fig. 4.7(a), (b), and (c) is calculated. The relationship between fractal dimension and RMS roughness follows a square-root law\cite{14} as in Eq. 4.13. $a$ is a constant of proportionality whose value is calculated from the slope of the lines in Fig. 4.14. Since this relationship is so simple, it implies that appropriate normal distribution PMFs (i.e., error functions) can be simply calculated given only a fractal generator and fractal dimension desired. Essentially, fractal surfaces can be replaced with their normal approximations. This has been implemented in a preliminary way in F-TRIDYN.

$$RMS = a\sqrt{D}$$  \hspace{1cm} (4.13)

Fig. 4.15 shows an illustration of a fractal surface and its analogous statistical surface. Although the statistical surface does not have a well-defined, explicit morphology, it reproduces the effects of a rough surface by determining randomly which points are inside or outside of the surface based on an input distribution. For F-TRIDYN, the input distributions are based on RMS roughness, a measure which has been used to link the fractal and statistical models of surface Roughness.
Figure 4.1: Three examples of natural fractals that occur in the natural world. (a) Shows tungsten fuzz, a natural fractal created via Helium ion irradiation of Tungsten. (b) shows a member of the family Brassicae. (c) shows atomic-scale graphitized carbon.

Figure 4.2: An illustration of the first seven iterations of the Middle-Third Cantor Set.
Figure 4.3: The log-log fractal scaling law plot of a rasterized Koch Snowflake. The slope of the log-log fractal scaling law plot is shown. Inset is the rasterized Koch Snowflake used and its true, theoretical fractal dimension calculated using Eq. 4.2.
Figure 4.4: Log-log fractal scaling law plots for several coastlines and a simple circle[30]. Both the linear behavior of the fractal coastlines and the non-linear behavior of the non-fractal circle are readily apparent.
Figure 4.5: Fractal generators and their respective fractal dimensions calculated using Eq. 4.2[30]
Figure 4.6: A fractal generator, its encoding as a signed binary array, and a resulting fractal surface from 6 iterations using the fractal generator method.

Figure 4.7: Three examples of generators investigated for use in F-TRIDYN. (a) is the bounded triangular generator. (b) is the unbounded trapezoidal generator. (c) is the bounded trapezoidal generator.
Figure 4.8: A ray is cast upwards from the point P through an approximate fractal surface and the intersections that the ray makes with the surface are counted and labeled. This point is inside the surface, so the ray cast upwards from it (and indeed, any ray cast from it) will intersect with the surface 9 times.
Figure 4.9: A demonstration of the even-odd algorithm based on the Jordan Curve Theorem. In this demonstration, red Xs mark points, chosen at random, detected to be outside the surface. Blue Os mark points inside the surface.
Figure 4.10: A demonstration of PKA reconstruction using the stepping algorithm technique. Shown in the figure are a generated fractal, a randomized copy of that fractal, used to test the method, and the resulting reconstruction of the original fractal down to the smallest ruler length used. This method accurately measures fractal dimension to 3 significant figures, as accurate as the box counting method demonstrated in Fig. 4.3.

Figure 4.11: A histogram showing the height distribution of points of a generated fractal surface. The distribution of heights is related to the probability of a point at any depth being inside or outside of the surface.
Figure 4.12: Three examples of PMFs for a generated fractal, showing the probability of being inside the surface versus depth. (a) shows the PMF of a first-generation fractal. (b) shows the PMF of a second generation fractal. (c) shows the complex PMF of a sixth-generation fractal.

Figure 4.13: An illustration of measuring RMS roughness of a surface around a mean height, with $y_i$s represented by $r_1$ through $r_7$. 
Figure 4.14: A plot showing the proportionality constant $a$ from Eq. 4.13 for the three fractal generators in Fig. 4.7, along with a line showing how the RMS roughness scales versus fractal dimension. The RMS roughness axis, the $y$-axis, has been squared to show the linear dependence between the square-root of fractal dimension, $D$, and the RMS roughness itself.
Figure 4.15: An illustration showing a fractal surface, left, and a representation of its analogous statistical surface, right.
5 F-TRIDYN

5.1 F-TRIDYN Code and Libraries

Fractal-TRIDYN (F-TRIDYN) is a Monte Carlo, BCA code that includes explicit and statistical models of surface roughness, extensive input/output options for file-based code coupling, and an extensive set of supporting Python libraries that allow for the creation of fractal surfaces, the measurement of fractal dimension using multiple methods, generation of input files for the BCA code, and processing of output files. F-TRIDYN is based upon the older BCA code TRIDYN. F-TRIDYN’s added algorithms have been parallelized using OpenMP* and are more computationally efficient than previous implementations of fractal surfaces in BCA codes. F-TRIDYN consists of the following codes and supporting programs:

- F-TRIDYN
- Class for species parameters and lookup
- Class for simulation parameters
- Fractal surface class
- Fractal generation program
- Box dimension program
- Noisy surface analysis program
- Beam and target default simulation class

F-TRIDYN is the main code, written originally in Fortran 95 style. Additional algorithms have been written to be Fortran 2015 compliant. OpenMP* is used to handle parallelization through compiler directives. F-TRIDYN has been compiled using gfortran, the GNU Fortran compiler, and ifort, the Intel Fortran Compiler.

The species class holds the physical and experimental parameters for a particular species in the simulation. These include: mass, atomic number,
bulk binding energy, displacement energy, cutoff energy, amount of species in beam and target, atomic density in target, electronic stopping correction factor, incident energy, incident angle, maximum amount of species allowed in target, and the surface binding energy. These are the main parameters of the BCA model used in TRIDYN and F-TRIDYN[32]. These parameters fully describe the dynamics of collisions between any two species for a given energy and impact parameter. The energy is determined from the simulation dynamics and the impact parameter for each collision is chosen according to the material model in the code. The material model in TRIDYN (and subsequently F-TRIDYN) assumes an amorphous solid, so the impact parameter is chosen from a pseudorandom square-root distribution.

F-TRIDYN includes a species lookup table so that physical parameters for the simulation can be automatically taken from a table. The table used in F-TRIDYN is the table of physical parameters from SD.TRIM.SP and cites multiple sources[7][9][27], and since the BCA is relatively sensitive to several parameters, this can have a significant impact on the results of the code. Using machine-generated input files is thus much preferred to manual creation of files for use in F-TRIDYN.

Simparams is a Python class that holds various simulation parameters that control the function of the code outside of the particular species involved. These include the simulation name, the simulation number, number of histories, depth of the simulation, and other options. This class is useful for holding and subsequently extracting simulation parameters when used correctly in a Python script for running F-TRIDYN. Use of this class should assist organization of complicated variation of parameters for simulations investigating multiple interacting phenomena.

The fractal surface class holds the lists of $x$ and $y$ points that make up the explicit fractal surface in F-TRIDYN. Additionally, it contains methods for fractal generation and surface file output. Fractals are generated via the method described above in section 4.4. The generator chosen for F-TRIDYN is the bounded trapezoidal generator, for reasons outlined previously. Fractal generation in this library can produce any fractal composed of linear, equiangular components. Input parameters for this methods are $\beta$, the angle that the deviating line segments make with the horizontal, and a signed binary array of 0, −1, and +1 that is used to make the initial fractal generator referred to as the shape list. A zero in the shape list represents a horizontal line segment, +1 a line segment that deviates by $\beta$ upwards, and −1 a line segment that deviates by beta downwards. In order to remain volume conservative, the shape list must be rotationally symmetric around
All generators used in F-TRIDYN have this property in order to keep the mean surface height centered around 0. Additionally, the shape list must sum to zero in order for the end point to remain on the zero axis. This is necessary to properly represent the surface in F-TRIDYN. The fractal generator also has the capability to scale the surface to any length scale in angstroms. Commonly, the length scale is chosen such that the lattice parameter and the distance between points in the fractal surface are comparable, replicating the microstructure of the material. This parameter can play a large role in the effect of surface roughness on the sputtering yield, as is obvious when one considers the edge cases. First, consider the case wherein the entire surface is much smaller than the range of ions in the material. In this case, the surface will be effectively flat for the ion-solid interaction. On the other hand, if the size of the surface is much larger than the range of the ion, the roughness will no longer have an effect as only one line segment will have ions incident upon it, and it will behave as a flat surface at the local angle of that particular line segment.

In order to aid ease of use, several example functions are included in F-TRIDYN's python libraries. Beamandtarget is a Python function that produces an input file for a 2-species case of F-TRIDYN where the beam is composed solely of one species and the target of the other. This function includes reasonable default values for all parameters, such that simulations can be run with as little information as the chemical symbols of the two symbols desired, thanks to the species lookup method described above. Thus, an F-TRIDYN simulation can be generated and run in a matter of seconds, with virtually no work required.

In order to analyze the behavior of the BCA with uncertain input parameters, F-TRIDYN includes examples functions that iterate over various physical parameters for either target or beam species. For example, the surface binding energy is a value that varies in the literature. In order to quantify the uncertainty involved in using a parameter whose value with such a large confidence interval, the code can be run multiple times using changes in these parameters. This demonstrates the particular sensitivity of the BCA model using the interaction potential in F-TRIDYN to some of the physical parameters that are used as inputs.

5.2 F-TRIDYN Output Options

TRIDYN's main output file is designed to be human readable. It contains a recreation of the input file, the simulation options, deposited depths, areal
densities and fluences, and total counts of incident projectiles, backscattered projectiles, transmitted projectiles, and stopped projectiles, along with their energies (or range in the case of stopped particles). Sputtered particles and their average energies are also tracked. Frenkel pair creation is included at the end of the output file. Although this form of output file is useful for single simulations, it is not convenient for being read by computer, e.g. for code coupling. To this end, F-TRIDYN includes a concise output of particle counts in the form of lists for each simulation. These lists contain identifying characters that correspond to the quantity in question (B for backscattered, S for sputtered, etc.) and a number that indicates the species of the particles being counted that is identical to the component number in F-TRIDYN.

Angular distributions of sputtered and backscattered atoms are included in lists. These three angles are represented by directional cosines in x, y, and z. These distributions are exceptionally useful for coupling to plasma codes such as GITR[53], an impurity transport code for tokamaks. The angular distributions of sputtered and reflected particles will determine how these particles redeposit on the walls of a fusion reactor or other plasma device with coexisting electric and magnetic fields. These distributions are sensitive to a number of physical parameters, including surface roughness and the angle-of-incidence of the incoming ions, so a detailed understanding of the behavior of these distributions under changes to the system conditions is necessary for the design of future fusion and other plasma devices.

Figs. 5.1 (a)-(c) show example angular distributions of sputtered Tungsten atoms and reflected Helium atoms through Helium ion irradiation of a Tungsten target from F-TRIDYN. Alpha, Beta, and Gamma are the directional angles in the x, y, and z directions respectively, as defined in the TRI-DYN coordinate system[15]. These files form the basis of code-coupling to sheath and plasma codes such as GITR[53], which tracks impurity transport in tokamaks and other devices, including PISCES (a linear plasma device) and ITER. Specifically, these angular distributions were produced via 200 eV Helium ions incident on bulk tungsten at a normal angle of incidence.

The most useful output for coupling to materials codes such as ADR codes like Xolotl is the implantation profile. This is the distribution of stopped ions in the material. F-TRIDYN includes tracking of all stopped particles, and outputs them to lists for incident stopped ions, PKAs, and SKAs. These lists can be used to send information about ion implantation to codes that can then evolve those ions within the appropriate target material. The BCA does not include physics for thermal processes or processes outside the atomic time and length scales represented by the series of binary
collisions. For this reason, other codes must be used to track the evolution of chemically active species in targets. Helium in Tungsten is a particularly difficult system to model due to the complex interactions between Helium and itself, Helium and the Tungsten lattice, and Helium and vacancies and other forms of damage in the lattice. In Xolotl[37], these interactions are represented in an ADR code with rate constants measured from Molecular Dynamics simulations.

Fig. 5.2 shows an example implantation profile of 200 eV Helium ions incident upon pure tungsten. This is the quality of implantation profile from F-TRIDYN used to generate an implantation profile function (via high-order polynomial fit) as input for the ADR He-W material evolution code Xolotl. These implantation profiles are also of interest in other material modification fields.

Each of these components plays an important role in increasing ease of use of the F-TRIDYN code. SRIM, one of the most popular BCA codes, is among the most used because of its ease of use. It includes a Graphical User Interface (GUI) that, combined with a large library of built-in physical parameters and compounds, makes it the easiest to use BCA code that is widely available. The development of the F-TRIDYN codebase has been pursued in order to make use of F-TRIDYN as simple if not more simple than SRIM. SRIM does not produce correct results in many situations[21]. Its continued popularity reflects the magnitude of the problem that difficulty of use represents even in scientific fields.

Additionally, F-TRIDYNs libraries have been expanded to ease code-coupling, as detailed in section 5.4. By itself, the BCA model does not reproduce many important PMI phenomena, including thermal effects, many important chemical and material effects, and more. Because the BCA model is effectively a zero-temperature model of the surface, many physical phenomena of interest especially to the fusion community require separate simulations to handle these effects. For example, the problem of tungsten fuzz growth initiation requires the use of timescales on the order of seconds, which is impossible in the framework of the BCA. For these reasons, F-TRIDYN libraries have been developed with ease of code coupling in mind, as well as ease of use for standalone simulations.

5.3 Sputtering Yield Simulations

F-TRIDYN has been used in an independent context (i.e., not coupled to other codes) to investigate the effects of surface roughness on plasma-
Figure 5.1: The angular distributions of sputtered atoms from an F-TRIDYN simulation in x, y, and z directional cosines.

Figure 5.2: An example implantation profile from F-TRIDYN, of 200 eV Helium ions incident upon a pure Tungsten target.
material interactions for systems relevant to both fusion energy and material processing. Specifically, three systems of relevance to the fusion field were investigated[14]. These were Argon on Tungsten, and Helium on Beryllium. First generation fusion power plants will use Deuterium and Tritium fuel and have Helium exhaust[17]. The D-T fusion reaction is reproduced in Eq 5.1. D-T fusion has the highest cross section at lower, more easily attainable energies and is thus the focus of first-generation fusion power plant designs.

\[ D + T \rightarrow \alpha + n + E \] (5.1)

For this reason, the effect of Hydrogen isotopes and Helium on common first-wall materials in fusion reactors is of utmost importance to the design of appropriate materials to handle the high heat and physical stresses of a fusion reactor environment. ITER, as designed, will have wall materials that include primarily Beryllium and Tungsten, hence their inclusion in this study[39]. The sputtering species for Tungsten in this study was chosen to be Argon for two reasons. First, argon one of the most commonly chosen gases used for sputtering due to its inclination to not react with most materials. Helium and Hydrogen may have complex interactions with the Tungsten lattice that are the focus of intense computational modeling beyond the abilities of a BCA code alone. Second, the sputtering threshold of Tungsten is relatively high (165 eV[20]), meaning that light-ion sputtering is difficult to perform to high statistics experimentally on a Tungsten target. For this reason, there is more experimental data Argon sputtering of Tungsten available, especially for variously roughened tungsten targets. Since the goal of F-TRIDYN is to include a fractal model of surface roughness, the more experimental data available to compare to drove the decision to simulate this system instead of the more fusion-relevant systems of He-W or H/D/T-W.

5.3.1 Helium on Beryllium

Energetic Helium produced via the D-T reaction in a fusion environment is of particular materials concern. Energetic Helium produced in this way may interact with the walls in a structurally negative way, introducing defects and modifying the surface structure[23]. Beryllium walls are planned for ITER[39]. Additionally, in preparation for ITER construction and operation, Beryllium walls were recently introduced in JET[22]. The introduction of Beryllium as a Plasma-Facing Component (PFC) material in the ITER-
like wall project degraded fusion performance. A significant part of the challenge of solving fusion lies in the problem of materials. Understanding why the wall material and wall topology may play such a significant role in fusion plasma performance is one of only a few as-of-yet unsolved problems in fusion. F-TRIDYN, by simulating the fast-timescale interaction between Helium and various wall materials, and by coupling with material and plasma codes such as Xolotl[37] and GITR[53] respectively, will provide a step towards shining a light on the detailed processes that lead from wall material to fusion plasma performance. A first step in investigating this phenomena is gathering information about sputtering yields. The sputtering yield can be used in place of a full BCA simulation as a wall reaction as long as the systems yield is simple enough to be reduced to a function of a single variable. To this end, sputtering yield calculations for rough and smooth Helium on Beryllium at varying energies have been included[14].

Fig. 5.3 compares the simulated sputtering yield of 300 eV Helium ions on Beryllium for flat and rough surfaces using F-TRIDYN. Comparison of these two curves, marked by the blue star and red circle, shows the marked reduction in sputtering yield for most angles of the rough surface. Additionally, significant sputtering yields are found at even very high angles due to the variation in surface height compared to the atomically smooth, flat surface. These two curves are compared to an angular model[26]. Their model used a semi-empirical technique for including the effect of surface roughness in the TRIM code. Rough surfaces were created and imaged using electron microscopy. These images were analyzed to calculate, at the position of the beam, a distribution of local angles of incidence for varying beam angle of incidence. The local angle of incidence was calculated using a 3D reconstruction of the surface from the SEM images. This method and F-TRIDYN are compared to experimental data. Sputtering yield experiments are notoriously difficult to perform, especially for rough surfaces, as indicated by the large spread in the three experimental values given for normal incidence[26]. Nevertheless, the trend of increased yield at low angles and decreased yield at moderate angles of incidence compared to the predicted sputtering yield of a flat surface is captured by F-TRIDYN. In order to estimate the fractal dimension, the fractal length scale of the target was taken from Küstner et al.’s measurements of the surface. They included an RMS measure of surface roughness. This was used to back-calculate the appropriate fractal dimension for the standard bounded trapezoidal generator and the length scale given to recreate a surface with the same RMS roughness as in the experimental work.
An advantage of F-TRIDYN over the local angle of incidence distribution method is that it is possible to simulate a rough surface theoretically, without the need to image and analyze a real, physical surface in order to produce useful results. This allows for one to simulate ion-solid interactions for a wide range of possible surface roughnesses without relying on semi-empirical calculations to determine what system to simulate. Removing the semi-empirical step in this process makes the fractal surface model significantly more flexible than the measured local angle-of-incidence model of surface roughness.

Fig. 5.4 shows the effect that fractal dimension has on 500 eV Helium incident on Beryllium at two angles of incidence. At this energy, the surface roughness does not play as significant a role as it would at a lower energy, since most of the relevant nuclear collisions are occurring well below the lowest surface depth, meaning that the effect of the surface is minimized. Nevertheless, at this incident ion energy, surface roughness does decrease the sputtering yield at 60 degrees for this system. On the other hand, the sputtering yield at normal incidence is increased. This effect reflects the multi-variate nature of the ion-solid interaction problem. Although at 500 eV the effect of surface roughness is relatively small, at lower energies it plays a significant role, as seen in the change in the angular dependence on sputtering yield in Fig. 5.6.

Fig. 5.5 shows the energy dependence of the sputtering yield of Helium on Beryllium at normal incidence for three surface roughness values, with fractal dimensions (FD) equal to 2.0 (smooth), 2.1, and 2.3. This plot is shown on a log-log axis to show that the high speed performance of F-TRIDYN allows for the practical calculation of sputtering yields at near-threshold energies. Here, the sputtering threshold of Helium on Beryllium is found to be approximate 20 eV. The sputtering threshold is slightly decreased with increasing surface roughness. As the incident ion energy is increased, the yield increases dramatically, from 10E-3 to nearly 10 atoms/ion. At the energy of maximum yield the effect of increasing surface roughness is most apparent. Each increase in surface roughness significantly increases the yield of Helium on Beryllium at normal incidence. This is important because, as a fusion reactor runs to longer operation times, the Beryllium surface will be modified by the incident ions and other energetic particles coming from the fusion plasma. These particles may, depending on the species, angle of incidence, and energy of incidence, roughen or otherwise modify the surface morphology. If Beryllium impurities lead to a significant decrease in fusion plasma performance, as the evidence from JET suggests[22], then as surface
Figure 5.3: A plot comparing F-TRIDYN results of rough and smooth surfaces to experiments and simulation results using the alternative local angle-of-incidence model of surface roughness from [25]. F-TRIDYN results are shown for a smooth (fractal dimension=2.0) surface and a rough surface whose fractal dimension was chosen to match the RMS roughness observed in the experiment plotted.
Figure 5.4: A plot showing the effect of increasing surface roughness on sputtering yields of Helium on Beryllium at 500 eV for normal incidence and 60 degrees.
modification continues, fusion plasma performance may continually worsen. It is not feasible to replace every wall tile in a fusion reactor after every tokamak shot, so materials should be picked that will tend to smoothen under fusion conditions instead of roughen to avoid the negative implications of this effect.

Fig. 5.6 shows the angular dependence of Helium on Beryllium sputtering yields at 500 eV for smooth and slightly rough (FD=2.1) surfaces. The effect of increasing surface roughness is different in the two regions of the angular sputtering yield curve. In the low-angle-of-incidence region of increasing yield vs angle, the effect of surface roughness is slight. At low angles the effect of increasing surface roughness is to increase the sputtering yield. At higher angles, the effect of increasing surface roughness is to decrease the sputtering yield. This second effect is caused by a shift to the right of the angle of maximum sputtering yield, seen in theoretical calculations[50] and alternative computational surface roughness models[26]. This effect is especially significant in fields where sputtering is either very desirable or very undesirable. As an example of the first field, sputter deposition guns can be engineered to maximize sputtering yield of the target by choosing the angle of maximum sputtering yield for that system. As the target erodes and roughens, however, the angle of maximum sputtering yield will increase, and the system performance will decline. On the other side of this problem, the angle of incidence of the most energetic particles in fusion reactors is chosen to be very close to parallel to the surface in order to minimize the sputtering of the walls. If the angle of maximum yield is shifted to the right as the surface is roughened under fusion relevant conditions, this will significantly increase sputtering yield from the wall and significantly affect fusion plasma performance via impurity losses.

5.3.2 Deuterium on Beryllium

Deuterium is the most common analogue gas used for simulating fusion environments. Reasons for this choice are twofold. First, Deuterium and Tritium, the two fuel gases for a first-generation fusion reactor such as ITER, are both isotopes of hydrogen. Chemically they are almost identical. Physically, the only difference in their dynamics is their mass. Because of this, Deuterium is not only an actual fusion fuel but also a relatively close analogue to Tritium for simulating fusion-relevant environments. Second, Tritium is both radioactive and expensive. It is often too expensive and difficult to use safely to use in simulating fusion environments. Although it will be
Figure 5.5: A plot showing the log-log dependence on energy of the sputtering yield of Helium on Beryllium for three different surface roughnesses.
Figure 5.6: A plot showing the dependence of the sputtering yield on angle of incidence for a smooth surface (D=2.0) and a rough surface (D=2.1).
required for fusion events to occur, as the energy required for D-T fusion is significantly less than that of D-D fusion[17], Deuterium is a suitable substitute for preliminary experiments. For this reason Deuterium has been chosen as the fuel gas for calculating sputtering yields in F-TRIDYN.

As seen in Fig. 5.7, the effect of surface roughness on the sputtering yield of 500 eV Deuterium on Beryllium is more significant at high angles than that of Helium on Beryllium. Since fluxes of Helium and Deuterium coming from the plasma into the wall will differ significantly, it is important to understand how the relative sputtering yields of the two species incident upon the same wall will change with surface roughness. As surface roughness increases, the sputtering yield of normal-incidence, 500 eV Deuterium on Beryllium is constant. Thus this system is stable even when the surface has significant surface roughness.

Fig. 5.8 shows the effect of incident ion energy on the sputtering yield of Deuterium on Beryllium. This system has a sputtering threshold of approximately 20 eV. Unlike the Helium on Beryllium system, increasing surface roughness has no effect on the sputtering threshold. This is another piece of evidence that D on Be is less sensitive to changes in surface roughness at low energy. This may be caused simply by the longer range of a lighter ion in bulk Beryllium, but is still worth investigating further. Since Deuterium will not have the same chemical effects with targets as Helium, it will be important to couple BCA codes to codes that handle longer timescales and non-zero temperatures, such as Xolotl[37] or MD codes, in order to fully understand the effect that these species have on fusion first wall materials. As surface roughness increases, a significant increase in sputtering yield is observed only near the energy of maximum sputtering yield. As the energy increase, the difference in the sputtering yield curves for the three surface roughness values presented (D=2.0, 2.1, and 2.3) converge. Since the most significant effect of surface roughness is observed at keV energies, it is less relevant to long-term operation of fusion reactors with the exception of Edge-Limited Modes, which may reach energies as high as these. In most other cases, the Deuterium on Beryllium system is less sensitive to changes in surface morphology than Helium on Beryllium.

Fig. 5.9 shows the effect of increasing surface roughness on the angular dependence of sputtering yield for the Deuterium on Beryllium system. As seen before, at low angles of incidence and moderate energies of incidence, this system is not sensitive to changes in surface roughness. On the other hand, the shift in angle of the peak of maximum sputtering yield is still observed. It is this effect that leads to the most significant change that
Figure 5.7: A plot showing the effect of increasing surface roughness on sputtering yields of Deuterium on Beryllium at 500 eV for normal incidence and 60 degrees.
Figure 5.8: A plot showing the log-log dependence on energy of the sputtering yield of Deuterium on Beryllium at normal incidence for three different surface roughnesses.
surface roughness has on this system. At very high angles of incidence, a smooth Beryllium target impacted by 500 eV Deuterium ions has very little sputtering. On the other hand, a slightly rough surface, with FD=2.1, has a sputtering yield near the maximum for this system out to 89 degree angles of incidence. This is of great import for fusion reactor design. Often, magnetic field lines that drive ions to very high angles of incidence are used to control the sputtering yield of PFCs in fusion reactors. The Deuterium on Beryllium system, if it is roughened by the fusion environment, will begin to sputter significantly despite this design choice. This is one reason why detailed investigation of the effect of surface roughness on PMI is necessary. Only at very high angles, which are hard to design in an experimental context, is this effect apparent. For this reason, F-TRIDYN serves to perform an informative role in showing that rough Beryllium will still exhibit high sputtering yield by energetic Deuterium even at very high angles of incidence.

5.3.3 Argon on Tungsten

Argon is among the most commonly used gases for sputtering because it is relatively massive, is relatively cheap, and, in most situations, is nonreactive. These three properties make it a very attractive gas especially for sputtering metals. Argon is a gas traditionally used for sputter deposition of metals for these reasons[28]. Tungsten is an important target material for fusion reactors because it has one of the highest sputtering thresholds of any widely available metal, around 200 eV for light-ion sputtering. Also, Tungsten also has the appropriate material strength to be used as a structural component in fusion reactors. For these reasons many early fusion reactor designs called for all-Tungsten walls. Unfortunately, the interaction of energetic Helium with Tungsten was discovered to be significant. Particularly, Helium ions incident upon bulk tungsten caused, at certain fluences, energies, and substrate temperatures, unprecedented changes to surface morphology, collectively referred to as Tungsten Fuzz. Because this process is driven by a combination of thermal and chemical processes, the BCA alone cannot simulate it. However, the BCA can provide a link between plasma and material codes such that tungsten fuzz can be simulated at large length and time scales. Due to this significant interaction between fusion-relevant fuel and exhaust species and Tungsten, and due to available data for Argon sputtering of metals for the reasons outlined at the beginning of this section, for the pure Tungsten sputtering yield results in this chapter the nonreactive
Figure 5.9: A plot showing the dependence of the sputtering yield of Deuterium on Beryllium at 500 eV on angle of incidence for a smooth surface (D=2.0) and a rough surface (D=2.1).
Argon was used.

Fig. 5.10 shows the effect of surface roughness on sputtering yields at normal incidence and a 60-degree angle of incidence for Argon ions on a Tungsten target at 500 eV. This combination showed a significant effect of surface roughness on sputtering yield for 60-degree angles of incidence. Increasing surface roughness at this angle resulted in a significant decrease in sputtering yield, from approximately 0.7 atom/ion to below 0.6 atom/ion. It is likely that Tungsten PFCs in fusion reactors will experience significant surface roughening due to ion bombardment, especially by energetic Helium ion bombardment. For this reason, the decrease in sputtering yield as surface roughness increases is actually a positive sign that the surface roughness will not have an entirely degrading effect on fusion plasma performance. On the other hand, the introduction of Tungsten fuzz to the surface of PFCs may lead to macroscopic erosion. Tungsten fuzz nanostructures are incredibly fragile and can be damaged by a light touch. Because of this effect, fuzzy Tungsten may be a significant source of high-Z dust in the plasma, which could potentially have a disastrous effect on fusion plasma performance. This illustrates a scenario where the dynamics of surface morphology must be understood so that they can be controlled, leading to a rough surface that sputters less but is not as fragile as long Tungsten fuzz nanostructures.

Fig. 5.11 shows the energy dependence of Argon on Tungsten sputtering yields at normal incidence for three surface roughnesses. This figure shows the importance of the angular dependence on sputtering yield. At normal incidence, there is no significant effect on sputtering yield from increasing surface roughness at any energy. The sputtering threshold remains the same for all three roughnesses. This means that at normal incidence, there is no way to modify a Tungsten surface to increase the sputtering yield for sputter deposition. If one desires to perform sputter deposition with Tungsten targets, higher angles of incidence are required to take advantage of changes to sputtering yield caused by surface morphology in either direction.

Although surface roughness has little effect on the energy dependence of the Argon on Tungsten system at normal incidence, at off-normal angles of incidence the story changes. Fig. 5.12 shows the dramatic effect that increasing surface roughness has on the sputtering yield of Argon on Tungsten at moderate energy (500 eV). Similar effects of increasing surface roughness on the angular dependence of sputtering yield are observed in this system to other systems, such as Helium on Beryllium and Deuterium on Beryllium. First, in the low-angle regime, the sputtering yield is decreased at all angles of incidence except normal. At high angles of incidence, however, increasing
Figure 5.10: A plot showing the effect of increasing surface roughness on sputtering yields of Argon on Tungsten at 500 eV for normal incidence and 60 degrees.
Figure 5.11: A plot showing the log-log dependence on energy of the sputtering yield of Argon on Tungsten at normal incidence for three different surface roughnesses.
the surface roughness leads to a dramatic increase in the sputtering yield due to the peak-shifting effect discussed in the previous two sections and seen in other works\cite{50}\cite{26}. Because Tungsten tiles are planned for use as divertor tiles in ITER\cite{39}, this may prove to have a significant effect on fusion plasma performance as the Tungsten tiles undergo morphology change from incident Helium ions. At even moderate surface roughness, at high angles of incidence the sputtering yield remains very high, at least for Argon on Tungsten. Self-sputtering may play a particularly important role here, as that would be the kind of heavy-ion sputtering that this figure can serve as an analogue for. Coupling to material codes will be necessary to understand the full effect of surface roughness on sputtering yields from evolving Tungsten due to implanted Helium ions, and this is the focus of section 5.4 on code-coupling to Xolotl.

Sputtering is a very important PMI process. Sputtering is of economic and scientific importance in a number of fields, including material processing, chip manufacturing, and fusion energy. Multiple strategies for modeling sputtering have been developed, including sputtering yield formulae and BCA and MD simulations. F-TRIDYN is a fast, accurate BCA code that can be used to simulate sputtering yields for a wide variety of complex materials. F-TRIDYN includes both a fractal and statistical model of surface roughness, which plays a significant role in sputtering yields. Surface roughness and target composition are difficult to include in sputtering yield formulae, motivating further development of BCA codes for situations where yield formulae are not sufficient. A primary use case for F-TRIDYN is the Hydrogen-Helium-Tungsten system, which is a fundamental part of a whole-device model for ITER. ITERs tungsten divertors require adequate study before operation due to the complex morphological effects that Helium induces in bulk Tungsten known as tungsten fuzz. Especially important will be the effect that Helium and Hydrogen have on sputtering yields from the Tungsten tiles, since Tungsten impurities may have a catastrophic effect on fusion plasma performance. For other systems, F-TRIDYNs sputtering yield results can be used in a table-lookup form to act as a material-plasma interface. F-TRIDYN is fast enough to produce highly detailed lookup tables with multiple independent variables to support dynamic PMI simulations.

5.4 Surface Roughness Evolution in F-TRIDYN

Interactions of charged, energetic Beryllium on Tungsten will be an important process in ITER. ITERs current design calls for Beryllium walls and
Figure 5.12: A plot showing the dependence of the sputtering yield of Argon on Tungsten at 500 eV on angle of incidence for a smooth surface (D=2.0) and a rough surface (D=2.1).
a Tungsten divertor[39]. Beryllium was chosen because of its relatively low atomic number and its material properties. Tungsten was chosen for its very high sputtering threshold. It is now known that Tungsten undergoes deleterious morphological effects when exposed to a Helium plasma at a certain flux and temperature. This process is known as tungsten fuzz. Although Tungsten fuzz is not caused by purely ion-driven atomic motion, the fast processes of Beryllium ions incident on Tungsten are. Thus Beryllium, sputtered or otherwise emitted from the wall, may become highly charged and gain significant energy in the fusion plasma environment. Since Beryllium is significantly heavier than fusion fuel products, it may play a kinetic role in sputtering from Tungsten or modifying the surface of the divertor.

Fig. 5.13 shows how a Tungsten surface evolves in Fractal Dimension when impacted with normal incidence, 100 eV Beryllium ions. In all surface evolution simulations conducted thus far, surface roughness equilibria have been found. In the case of Beryllium on Tungsten, two equilibria have been established in the neighborhood of low surface roughness. Any surface that begins smooth or with a fractal dimension below approximately 1.015 will tend towards 1.015. Any surface that is rougher will tend towards a fractal dimension of 1.045. These equilibria point to some competing processes, but the exact physics are not yet understood.

Fig. 5.14 shows the same system at higher roughness. At significant fractal dimension, all reasonable initial fractal dimensions converge towards approximately 1.125. A characteristic error bar of this process is shown in the figure. Because this equilibrium is approached by all initial fractal dimensions above the very smooth cases covered in Fig. 5.13, it may represent a more significant effect of the competing processes involved in surface evolution driven by ion bombardment.

Another significant system that exhibits surface evolution is Argon ions incident upon Silicon. Most silicon targets are crystalline, so the following results do not apply to pure crystalline silicon. On the other hand, polycrystalline Silicon is approximately amorphous and these results may be relevant for that case.

Fig. 5.15 shows how a silicon surface evolves from perfectly smooth to be rough with incident Argon ions at 100 eV at normal incidence. Starting from a fractal dimensions of 1.0, the surface evolves in a decaying exponential fashion to a rougher surface. There is some local variation from generation to generation, but the trend is relatively smooth, suggesting that this is an appropriate system for fitting to an empirical curve. An empirical fit to the simulation results has been developed, of the form shown in Eq. 5.2.
Figure 5.13: A plot showing how surface roughness of relatively smooth Tungsten targets changes with incident Beryllium ions. Each line represents a set of 20 simulations run one after another from different initial conditions of surface roughness.
\[ A \mp (A - 1) \exp (-C\Phi) \] (5.2)

Where A and C are constants of fit and \( \Phi \) is the fluence. This fit matches the simulation results and predicts a saturation roughness of the variable A in Eq. 5.2. Future work on determining what controls the constant A may allow one to predict how a surface will respond in terms of surface roughening or smoothing to ion irradiation. This knowledge is important both in fields where roughening is desired and where it is not. For example, in bioengineering, roughened surfaces are routinely used as substrates for biological material. If the surface roughness of a particular ion/target combination could be predicted, then appropriate materials and processing gases could be chosen to achieve a desired degree of surface roughness. On the other hand, in fusion engineering, surface roughness can introduce undesired material weakness. Modeling this interaction and determining what materials resist roughening when exposed to a fusion environment would valuable for sustained fusion performance over long tokamak or stellarator operational times.

Using the dynamic composition abilities of TRIDYN included in F-TRIDYN, Fig. 5.16 shows how the concentration of Beryllium evolves with increasing fluence as the surface morphology evolves. As fluence increases, the shape of the distribution of Tungsten concentration resolves into a curve with two distinct dips. The first dip represents the deposition of Beryllium on the surface of the Tungsten divertor tile. At the very surface, the concentration of Beryllium is near 1.0. The second dip most likely represents the results of self-sputtering of Beryllium back out of the material or the effect of increasing surface roughness changing the implantation depth, and therefore deposited energy distribution, of the incident Beryllium ions. This phenomena requires more investigation. Implementation of the full tracking of all sputtered atoms, including their initial positions, will aid in the understanding of this proces.

Two possibilities for competing processes driving the fractal dimension towards various equilibria are the introduction of RMS noise solely through the random process of collision cascades and the averaging effect of having many incident ions. When the two processes equalize, one of the equilibria will occur.

To understand fully the processes at play, future investigation into the roles that the choice of length scales, surface models, and more have on modeling evolving surfaces in a BCA code. As a next step, recreating the
Figure 5.14: A plot showing the evolution of fractal dimension of a rough Tungsten surface after 40 subsequent simulations of 100 eV incident Beryllium ions.

Figure 5.15: A plot of the evolution of a silicon surface to steady state roughness under bombardment of 100 eV, normal incidence Argon ions.
results of the fractal model using the statistical surface model described in section 3.2.4 may be able to give insight into the physics of this process. Since the statistical model recreates sputtering yield results of the fractal model, it should serve as a check on the assumptions used when fractal surfaces are reconstructed for surface evolution purposes.

Surface evolution in a BCA code has been achieved. Questions remain about the most correct way to include the required steps in a larger code. F-TRIDYN includes algorithms for the output of PKAs and the reconstruction of resulting atomic surfaces therefrom. Results of this set of algorithms working in concert to iteratively run BCA simulations on evolving surfaces have been promising. They show multiple equilibria in surface roughness evolution, indicating that competing processes may lead to saturation of surface roughness based on fluence upon ion irradiation. If these processes could be understood, at least in situations where the BCA surface evolution model remains valid, it would pave the way towards a greater understanding of how PMI affects surfaces and how the surface response including roughening and smoothing plays a role in the entire system.

Future work is necessary to determine regimes of validity for this model. Theoretical work on the PKA positions and whether they are an appropriate approximation of surfaces resulting from ion bombardment is needed. Determining what surface models are appropriate for each system will be very important, as the fractal model may not be accurate for different species of target and beam.

An alternative method of including surface evolution in a BCA code may be to couple the code to a surface evolution module that calculates the local free energy of the PKA motion in order to find a resultant surface instead of reconstructing and measuring a surface directly from final PKA positions. This alternative method may be more accurate for systems that do not meet the standards of validity for the pure BCA surface evolution model, and in fact may be more accurate for all systems. Development of this alternative model is currently not planned for F-TRIDYN, but advances in BCA code coupling including the inclusion of more input and output modes in F-TRIDYN allow it to be a platform for this and other file-based code coupling projects with minimal work.

5.5 Code Coupling

A primary goal of F-TRIDYN is to make the code available for use in code-coupling situations. F-TRIDYN has been coupled to both plasma and mate-
rial codes. On the plasma side, F-TRIDYN allows as input incident energy distributions and mixed composition of incident ion species. F-TRIDYN tracks particles that sputter, reflect, implant, or transmit in the surface as well. On the materials side, these tracked particles can be used to model fast ion processes so that thermal and other, slower processes can be handled by a different code.

As part of the PSI-SciDAC project, F-TRIDYN has been coupled in a preliminary fashion to the codes Xolotl[37] and GITR[53]. Xolotl is an ADR material code that tracks the motion and reactions of impurities and defects in a Tungsten lattice. Implanted ions, whose distribution profiles are created with an F-TRIDYN simulation, are allowed to evolve in the lattice. Damage and sputtered particles are also tracked using F-TRIDYN simulations. GITR is an impurity transport code that uses F-TRIDYN to handle the response of the surface under plasma contact. Sputtered target atoms are tracked as impurities in the plasma.

Coupling to both plasma and surface codes allows F-TRIDYN to bridge the gap between processes in the plasma and processes in the surface. These processes interact in a complex way that is not easily handled by analytic methods. Bridging this gap brings about the possibility of whole-device modeling, allowing for the operation of a plasma device to be simulated at all relevant time and length scales simultaneously. Performing this task is a milestone for the PSI-SciDAC project. F-TRIDYN has been upgraded to include output files for file-based code coupling that TRIDYN does not have, including implanted ion tracking, PKA and SKA tracking, and more. Code-coupling improvements, including additional input and output options, will be a focus of future development of F-TRIDYN.
Figure 5.16: Dynamic composition over fluence of an evolving Tungsten target with 100 eV, normal incidence Beryllium ions bombarding it.
Understanding the complex interplay between plasma and surfaces is an important step to advancing the field of plasma physics. Plasma-Material Interactions comprise a rich and intricate set of processes that will play a significant role in the engineering of future plasma devices. Approaching PMI with simulations allows for the calculation of physical quantities unapproachable by analytical or theoretical approaches. The inherent mathematical complexity of the whole problem can be reduced to discrete processes whose relationships can be simulated explicitly. Using simulations to understand PMI is the next step forward to understanding the role that the surface will play in future plasma devices such as fusion power plants. Code-coupling provides a set of strategies for compartmentalizing the problem of PMI into separate domains with their own time and length scales, allowing for multiple, independent simulation methods to be linked. This paves the way toward simulating whole devices, from the material bulk to the surface, from the surface to the plasma sheath, and from the plasma sheath inward into the bulk of the plasma.

F-TRIDYN has been developed with code-coupling in mind. F-TRIDYN includes models of surface roughness, a physical parameter whose influence on PMI is not fully understood. As the interface between the plasma and the material bulk, F-TRIDYN captures the complex behavior of ion-material interactions with rough and variously composed surfaces. F-TRIDYN’s libraries and upgrades allow for simple use as an independent code and as part of a suite of codes coupled using file-based code-coupling procedures.

F-TRIDYN reproduces theoretical and experimental results on the effect of surface roughness on sputtering, among the most important ion-material interactions. These results show the power and practicality the fractal surface model offers. Work on an alternative, statistical model of surface roughness is promising and offers significant speed increases over the fractal surface model.
Future Work

Future work on F-TRIDYN will focus first on improving code-coupling for the PSI-SciDAC project funded by the U.S. Department of Energy. Additionally, the statistical model of surface roughness and its results and properties will be investigated. F-TRIDYN will also continue to be developed as an independent platform, with additional libraries to simplify ease-of-use of the code. Additional physics, such as improvements to the scattering model and inclusion of more accurate electronic stopping are also being considered, promising to potentially lower the minimum energy of validity of the BCA model. All of these efforts will continue to expand understanding of ion-material interactions, especially in a plasma environment such as in a fusion reactor. Engineering of fusion power plants and operation of experimental reactors such as ITER will rely on the results of whole-device models, and the inclusion of F-TRIDYN in those codes allows for more accurate and faster modeling of relevant surface interactions than alternatives such as MD codes or sputtering yield formulae.

Statistical surface models of surface roughness have already proved to be a fast, efficient alternative to explicit, morphological models. For this work, a normally distributed surface was chosen. Other distributions may yield more realistic results. Comparison of both fractal, non-fractal explicit, and various statistical models will provide crucial insight into which models are most useful for simulating PMI.

Additional possible areas of investigation include the reformulation of electronic stopping from first principles to attempt to capture low temperature behavior as well as high temperature behavior. A significant gap exists in atomic simulations of PMI between MD and BCA codes. There is no framework that captures behavior between the fully-detailed, atomic motions of MD codes and the binary, approximate collisions of BCA codes. Modification of the electronic stopping theory included in BCA codes to a more detailed model of electronic interaction may be the key in bridging the gap between BCA and MD codes. Lindhard-Shcarff theory is well founded and is useful for a wide range of incident ion energies, but when implemented in a BCA code it fails to capture the detailed electronic inter-
actions that happen at the surface for low energy incident ions. Low energy interactions are important to a number of physical processes, such as adsorption, desorption, and chemical reactions. Currently, the only framework for simulating these interactions is MD, which is computationally expensive, especially for complex systems that display long-range effects such as the Tungsten-Hydrogen-Helium system that drives the formation of Tungsten fuzz.

F-TRIDYN is currently not written to be fully parallelized, despite the readiness of the BCA framework to be fully parallelized. This is another potential area of work and would require the redevelopment of the base F-TRIDYN code in order to achieve full parallelization. If this can be accomplished, however, it promises orders of magnitude faster BCA codes in the future. Since BCA codes are Monte Carlo codes, each thread, representing the motion of one particle at a time, is independent of every other thread. Graphical Processor Units (GPUs) have thousands of cores, each of which could run an independent thread. Expansion of BCA codes to GPUs will be an important step in securing their place in the multi-code framework of HPC plasma codes.

An investigation of the experimental validity of the fractal surface model as compared to other models of surface roughness is also warranted. Using experiments, it will be possible to determine which model of surface roughness is most appropriate for a given system. Exploration of the effect of fractal dimension irrespective of the relevant length scale may also provide insight into the development of atomic scale structures via ion bombardment. Fractal theory remains a promising area of development for modeling many physical processes.

Modeling of surface evolution in a BCA framework has been achieved in a preliminary fashion in F-TRIDYN. This method has lead to interesting results, including the appearance of multiple equilibria of fractal dimension of a surface after ion bombardment. Surface modification via plasma bombardment remains an active area of research. If the fractal-BCA surface evolution model compares well to experiment, it may prove useful in approximating the broad effect of surface modification post ion-bombardment. In situations where PMI is used for large-scale effects, such as increasing surface roughness to improve the efficacy of a subsequent chemical reaction, the BCA-fractal surface evolution model may be used to choose ion energy and angle of incidence required to achieve the desired surface morphology.
References


