2D Feature Model - Mechanisms and Algorithms

Langmuir Project Documents

Wei Tian

October 27, 2020

Contents

Introduction

This document introduces and discusses the mechanims and algorithms for 2-D feature model.

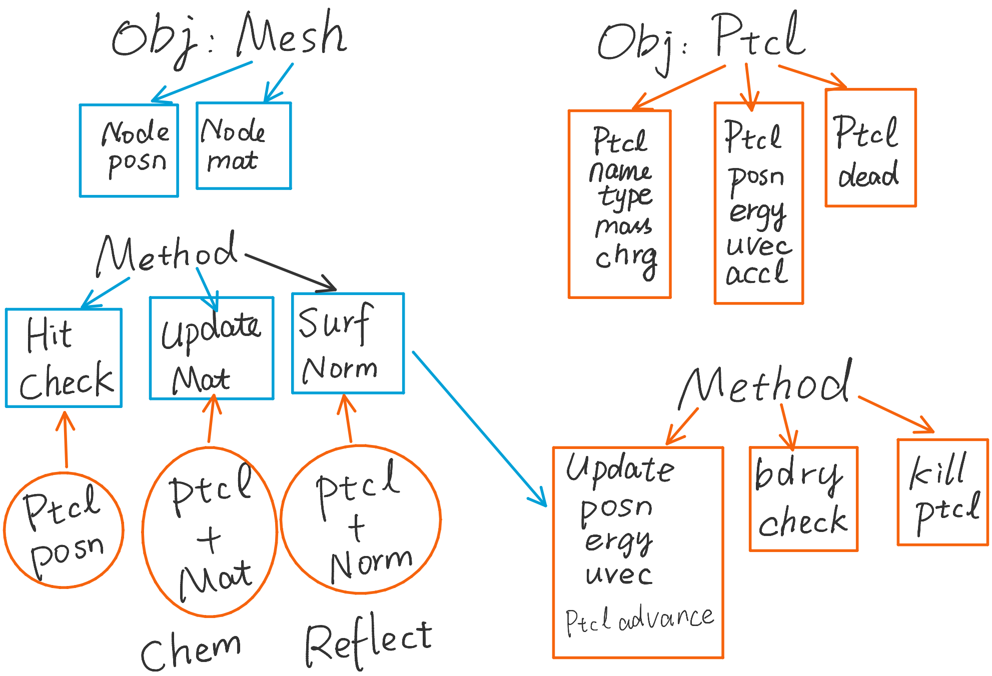
Feature Model 2D - Code Strucure

The feature model is designed following by an objective orentied programing. Four major objects are created: 1. Mesh 2. Particle 3. Reflect 4. React

Code Structure

Wednesday, September 2, 2020 3:25 PM

CODE STRUCTURE



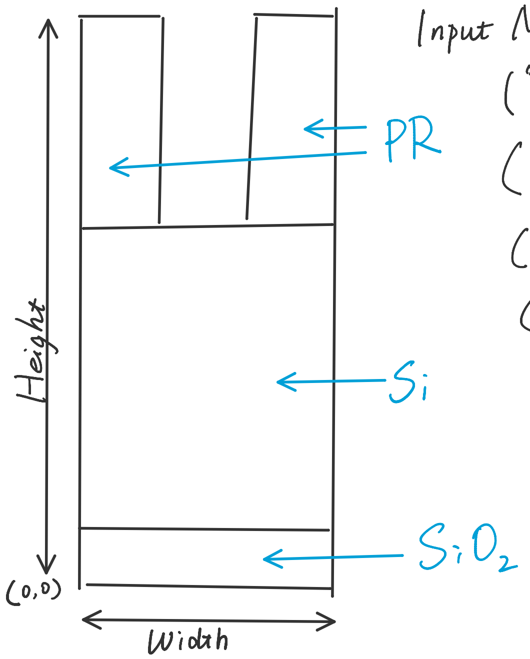
CHEM
REFLECT will be designed later

Feature Model 2D - Mesh Construction

The mesh is created by using voxel/cell structure. The voxel or cell center is defined and each represented a rectangular/square shape.

Mesh construction

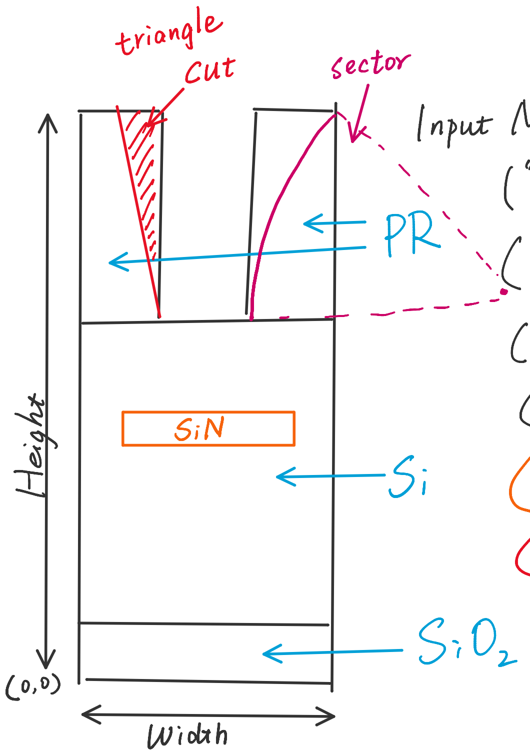
Sunday, September 13, 2020 9:44 AM



Input Mesh file:

("SiO_2", 1, "rect", 0, 0, 100, 50)
("Si", 2, "rect", 0, 50, 100, 300)
("PR", 3, "rect", 0, 350, 30, 100)
("PR", 3, "rect", 70, 350, 30, 100)

i) Utilize basic shapes to construct complex feature/geometry/mesh
ii) latter block/shape overwrites former
iii) "Vac" default as "0", used as "CUT"



Input Mesh file:

("SiO_2", 1, "rect", 0, 0, 100, 50)
("Si", 2, "rect", 0, 50, 100, 300)
("PR", 3, "rect", 0, 350, 30, 100)
("PR", 3, "rect", 70, 350, 30, 100)
("SiN", 4, "rect", 20, 250, 50, 10)
("Vac", 0, "tri", $\underbrace{30, 350}_{\text{Point 1}}$, $\underbrace{30, 450}_{\text{Point 2}}$, $\underbrace{20, 450}_{\text{Point 3}}$)
("PR", 3, $\underbrace{\text{"sec"}}_{\text{sector}}$, $\underbrace{180, 350}_{\text{Center}}$, $\underbrace{65}_{\text{radius}}$, $\underbrace{150, 180}_{\theta_1 \text{ to } \theta_2}$)

Feature Model 2D - Particle

Particles are created as an object, containing infomation consisting of constants and variables.

Particle

Sunday, September 13, 2020 10:21 AM

Particle	Name	Type	Mass(AMU)	Charge		Position	Energy(eV)	Direction	Dead
Arp	'Ar+'	Ion	40	+1		(0,0)	100	(-sin(α), cos(α))	0
O	'O'	Radical	16	0		var	var	var	var
E	'E'	Electron		-1		var	var	var	var

Constants

Variables

The Particle position is a computed to as the particle advances

Position_2 = position_1 + speed * direction * dt

Step = Speed * dt is fixed regardless of speed

The direction is a unit vector

The speed is not used. Instead the energy is used for reflection and reaction.

When reflection occurs, the direction and energy are updated

'Dead' is a tag to indicate whether the particle is alive or not

Dead = 0, continue to track the particle

Dead = 1, stop tracking the particle

Once a particle is created, the memory of the particle is a created and I will never be erased.

Two method:

1. Only a single particle memory is created, A new particle will use the same memory with updating all the constants and the variables
2. Since there are only a limit number of kinds of particles, multiple particle memories are created with updating only the variables

Feature Model 2D - Particle Tracing

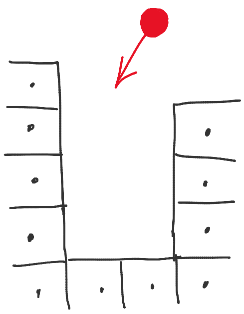
Particles are either moved by step advancing or tracked by ray tracing.
Boundary conditions are set to be either periodic or reflective.

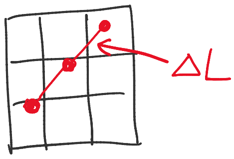
Step Advance

Wednesday, September 2, 2020 3:18 PM

STPE ADVANCE

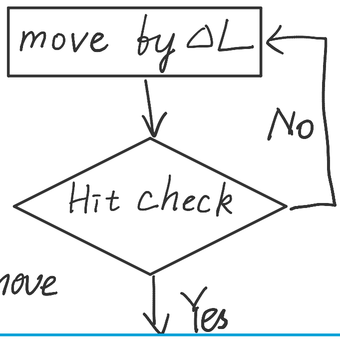
INFINITE SMALL PTCL

① 



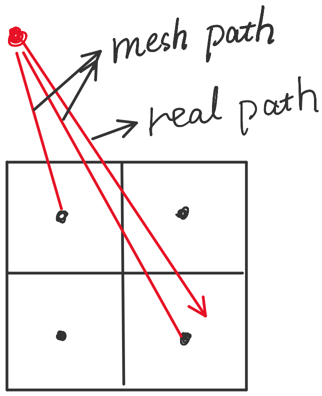
ptcl advances by step ΔL

Hit check after each move

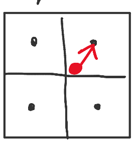


Computational Load depends on Mesh Domain

ptcl is always mapped to grids



Vac/Mat. ptcl (x_i, y_i)



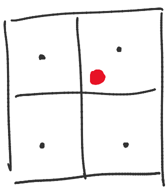
$i = \mathrm{round}(x_i/\Delta x)$

$j = \mathrm{round}(y_i/\Delta y)$

ptcl hits cell (i, j)

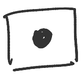
check cell to be vac. or Mat.

Two views :

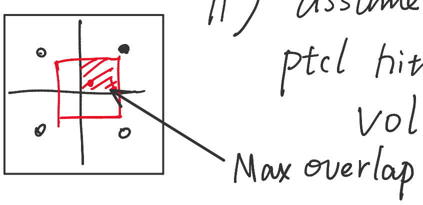


i) assume the ptcl is infinite small

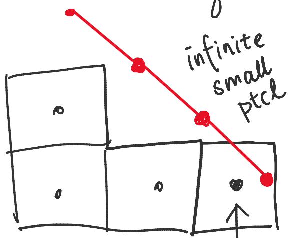
ptcl hits the cell when it enters inside a cell

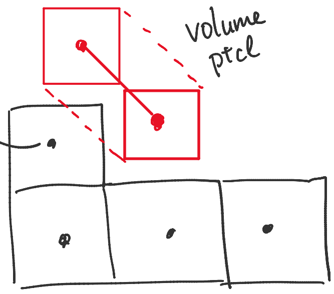
ii) assume ptcl has a volume 

ptcl hits the cell with max volume overlap



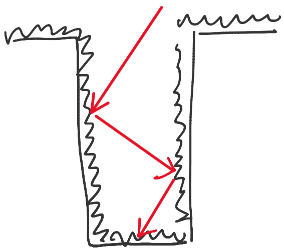
Possible questions:

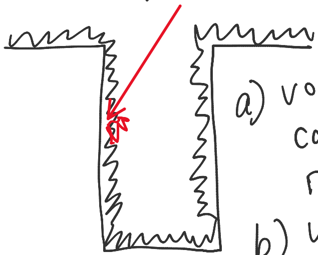




good for rough surf

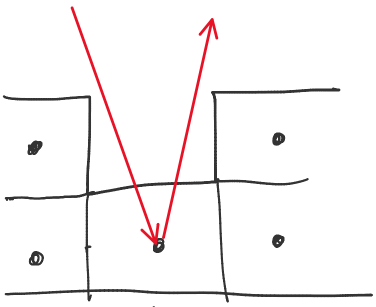
Trapped in rough surf

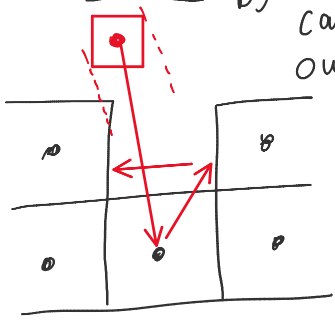




a) volume ptcl cannot reach narrow trench

b) volume ptcl cannot get out of narrow trench





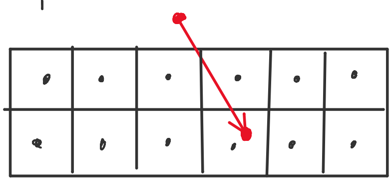
good for penetrating

$\Delta L = \Delta L(\varepsilon)$

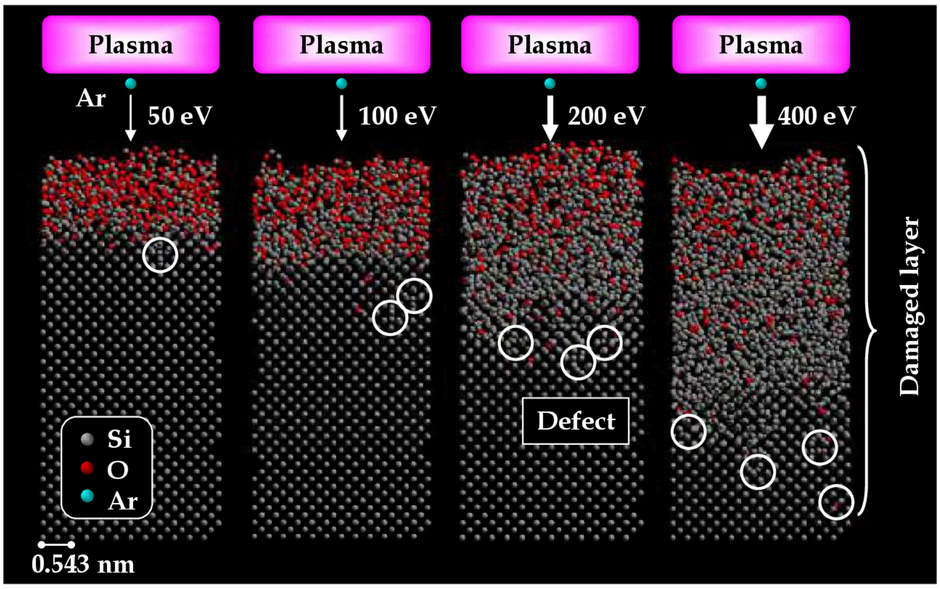
when use larger ΔL

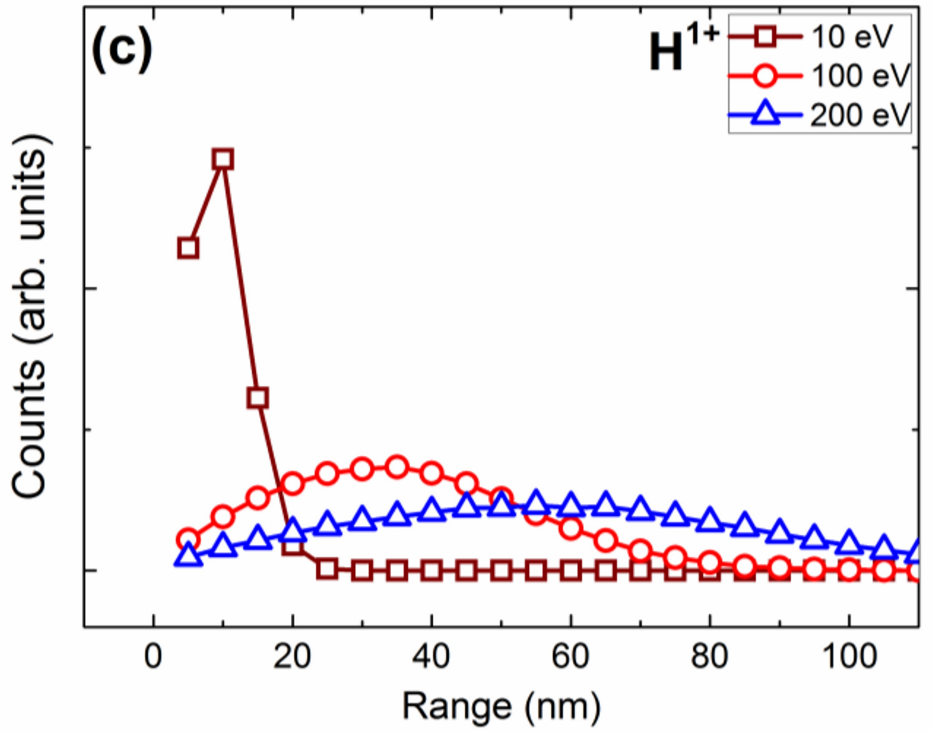
e.g. $\Delta L \approx 2\Delta x$

ptcl able to hit underlying layer



No need concept of SURF

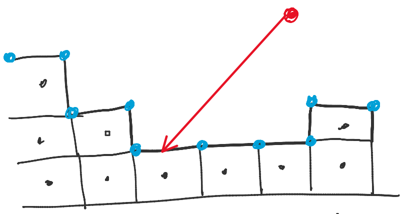

Plasma
Plasma
Plasma
Plasma
Ar
50 eV
100 eV
200 eV
400 eV
Damaged layer
Defect
Si
O
Ar
0.543 nm


(c)
H1+
10 eV
100 eV
200 eV
Counts (arb. units)
0
20
40
60
80
100
Range (nm)

Ray Tracing

Wednesday, September 2, 2020 3:19 PM

RAY TRACING



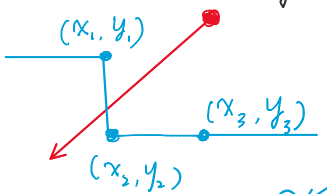
a) construct the surf
   surf nodes
   surf segments

surf can be paralled!

GPU 2D in para

b) sort the surf nodes

c) line: $ax + by + c = 0$



sign flips

$$\left.\begin{aligned} ax_1 + by_1 + c &> 0 \\ ax_2 + by_2 + c &< 0 \end{aligned}\right\}$$ Ray intersects with segment $(x_1, y_1) - (x_2, y_2)$

$$ax_3 + by_3 + c < 0$$

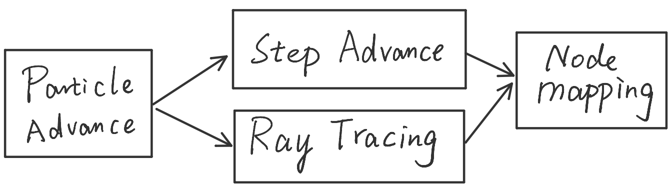
d) map segment $\longrightarrow$ Node/cell (i, j)

e) Ray intersects with cell(i, j)

Computational load depends on the surf area

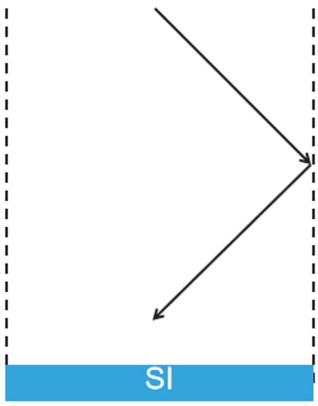
In 3D, computational load will be doubled ?

Eventually, all particles have to be mapped to meshgrid nodes. Surface intersection is a way of mapping. How to take advantage of RT ?

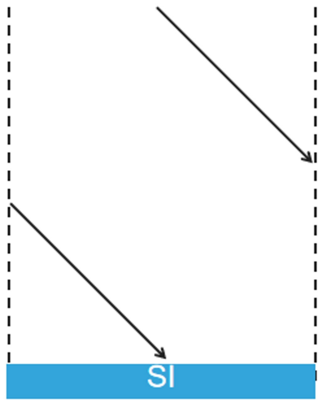


Boundary Condition

Sunday, September 13, 2020 8:15 PM



Reflective B.C.



Periodical B.C.

Feature Model 2D - Reflection

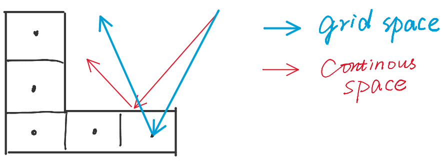
When a particle hits on a surface, there is a chance that the particle reflects from the surface.

Reflection

Wednesday, September 2, 2020 3:21 PM

REFLECTION

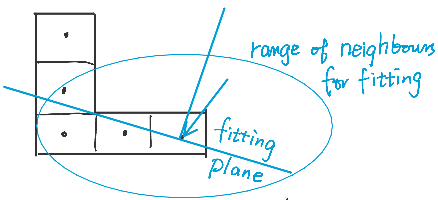
All Reflections occur on Nodes instead of SURF



Find the reflecting surf

Cell surf is only horizontal or Vertical

On nodes, surf is determined by plane fittin

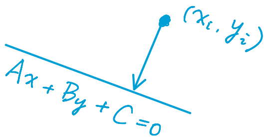


assume the fitting plane is

$$Ax + By + C = 0$$

for neighbours (x_i, y_i)

the distance from (x_i, y_i) to plance



$$dist = \frac{|Ax_i + By_i + C|}{\sqrt{A^2 + B^2}}$$

minimize total dist

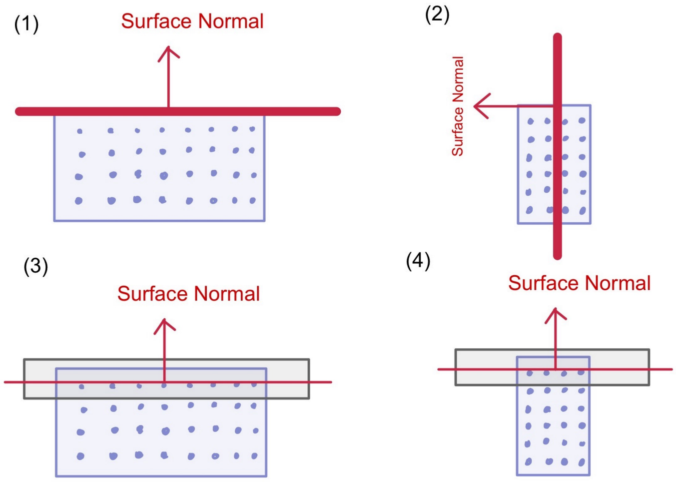
$$D = \sum_i dist(x_i, y_i)$$

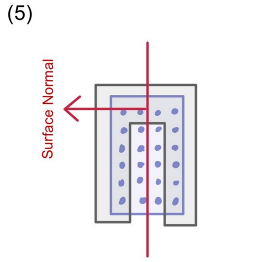
Usually, a simpler form is used

$$Q = \sum_i (Ax_i + By_i + C)^2$$

This sum can be only fitted on nodes

Once the fitting plane is detemined





1. It is better to use surface nodes than volume nodes.

2. The radius has an impact of surface normal calculation.

This surf can be only fitted on nodes
Once the fitting plane is determined
surf norm $\hat{n}$ is chosen from
Mat to Vac

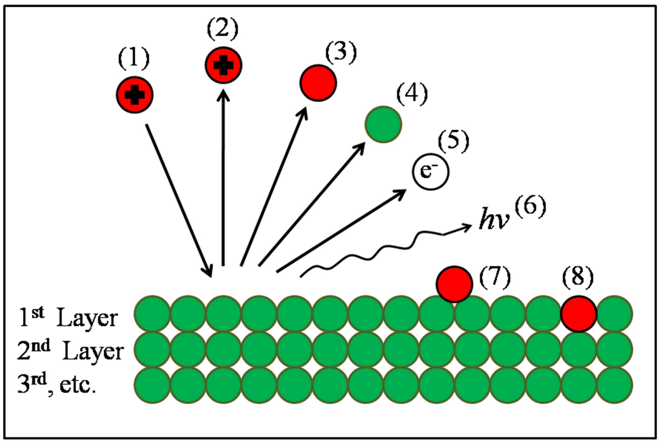
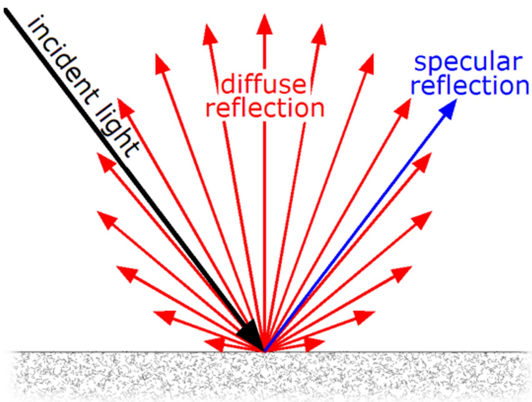
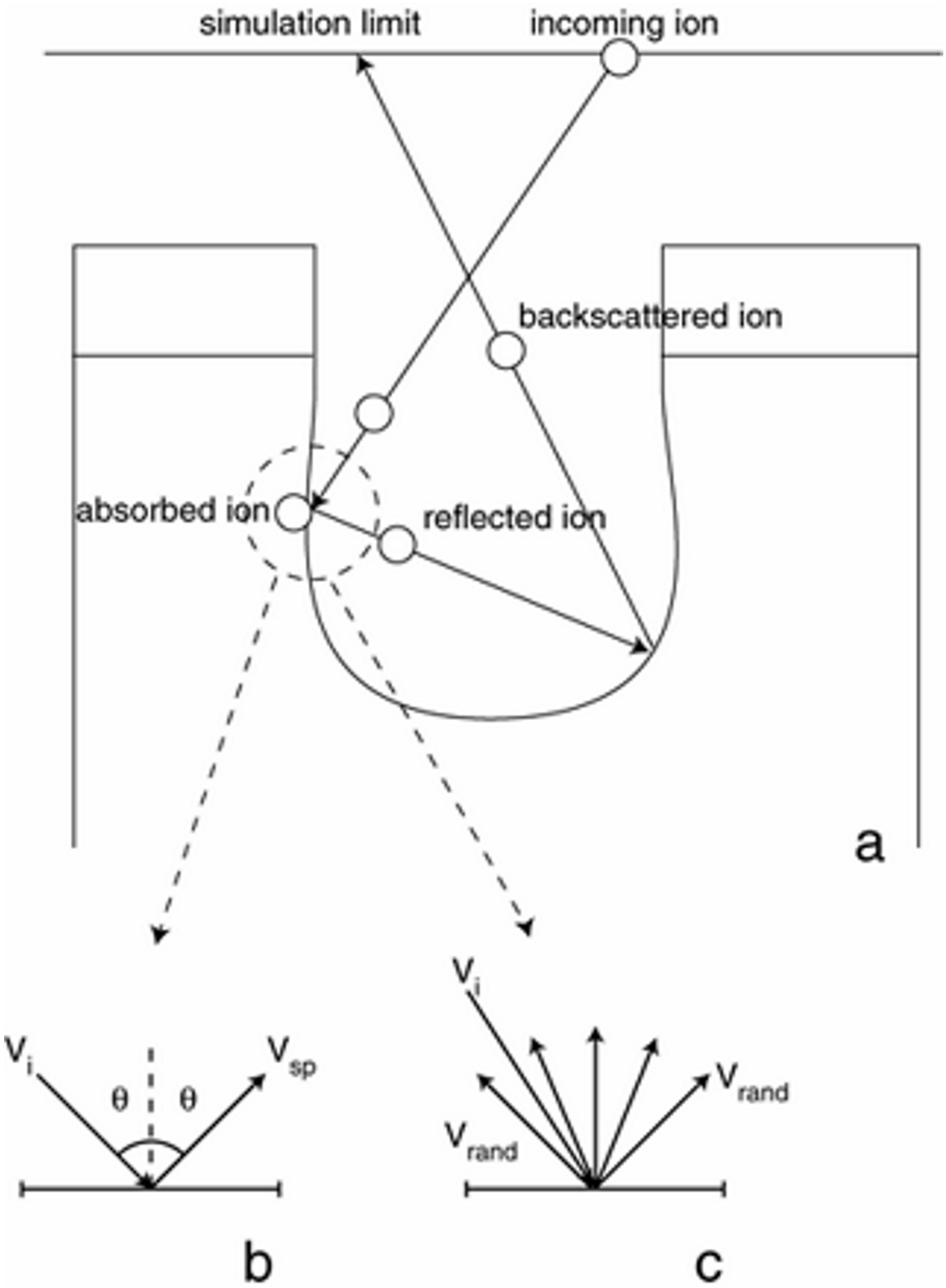


Diagram of various ion-surface interactions (non-exhaustive). (1) Incoming ion; (2) Scattering; (3) Neutralization and scattering; (4) Sputtering or recoiling; (5) Electron emission; (6) Photon emission; (7) Adsorption; (8) Displacement.

DIFFUSE REFLECTION

- Neutral particles with thermal, or near thermal (< 1 eV) energies reflect or re-emit from surfaces diffusively following a cosine angular distribution.
- This occurs because each particle is in thermal equilibrium with the surface, allowing them to briefly physisorb to the surface, before being re-emitted into the gas by vibrational processes in the solid, such as phonon scattering.
- The emitting probability density function depends on the particel and surface condition.





simulation limit
incoming ion
backscattered ion
absorbed ion
reflected ion
a
V_i
θ
θ
V_sp
b
V_i
V_rand
V_rand
c

ENERGY LOSS DURING THE REFLECTION

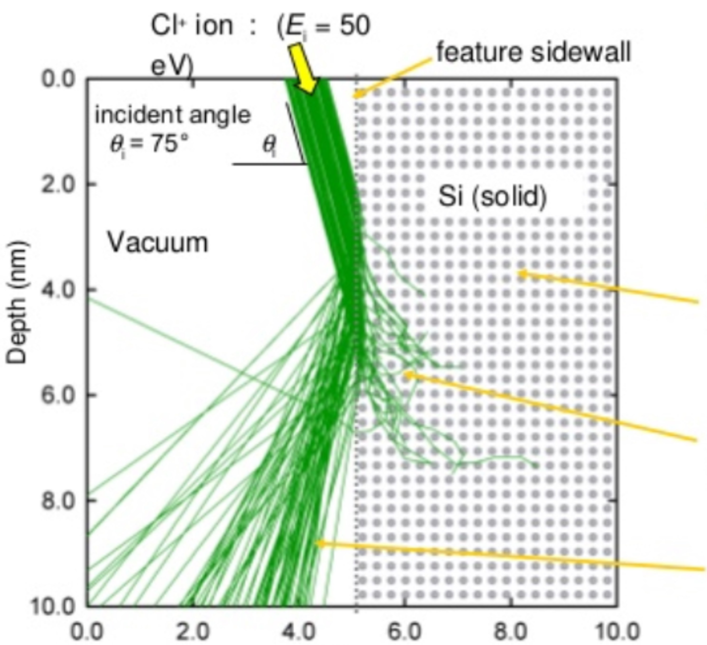
- Particles lose energy during the reflection as a function of initial energy and incident angle.

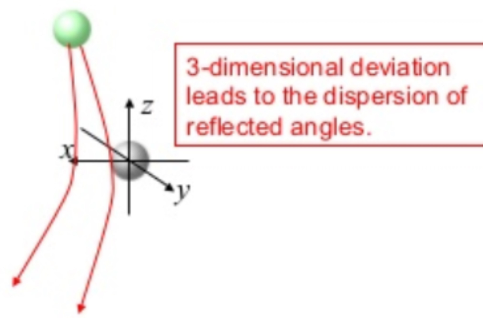
$$\varepsilon_f = \gamma_0 f(\theta) f(\varepsilon_i)\varepsilon_i$$

$$f(\varepsilon) = \begin{cases} 0 & \text{if} \quad \varepsilon < \varepsilon_0 \\ \dfrac{\varepsilon - \varepsilon_0}{\varepsilon_s - \varepsilon_0} & \text{if} \quad \varepsilon_0 \le \varepsilon \le \varepsilon_s \\ 1 & \text{if} \quad \varepsilon > \varepsilon_s \end{cases} \qquad f(\theta) = \begin{cases} 0 & \text{if} \quad \theta < \theta_0 \\ \dfrac{\theta - \theta_0}{\theta_0} & \text{if} \quad \theta > \theta_0 \end{cases}$$

for example, $\gamma_0 = 0.85$, $\varepsilon_0 = 0$ eV, $\varepsilon_s = 50$ eV and $\theta_0 = 30°$

Sample trajectories of reflected ions

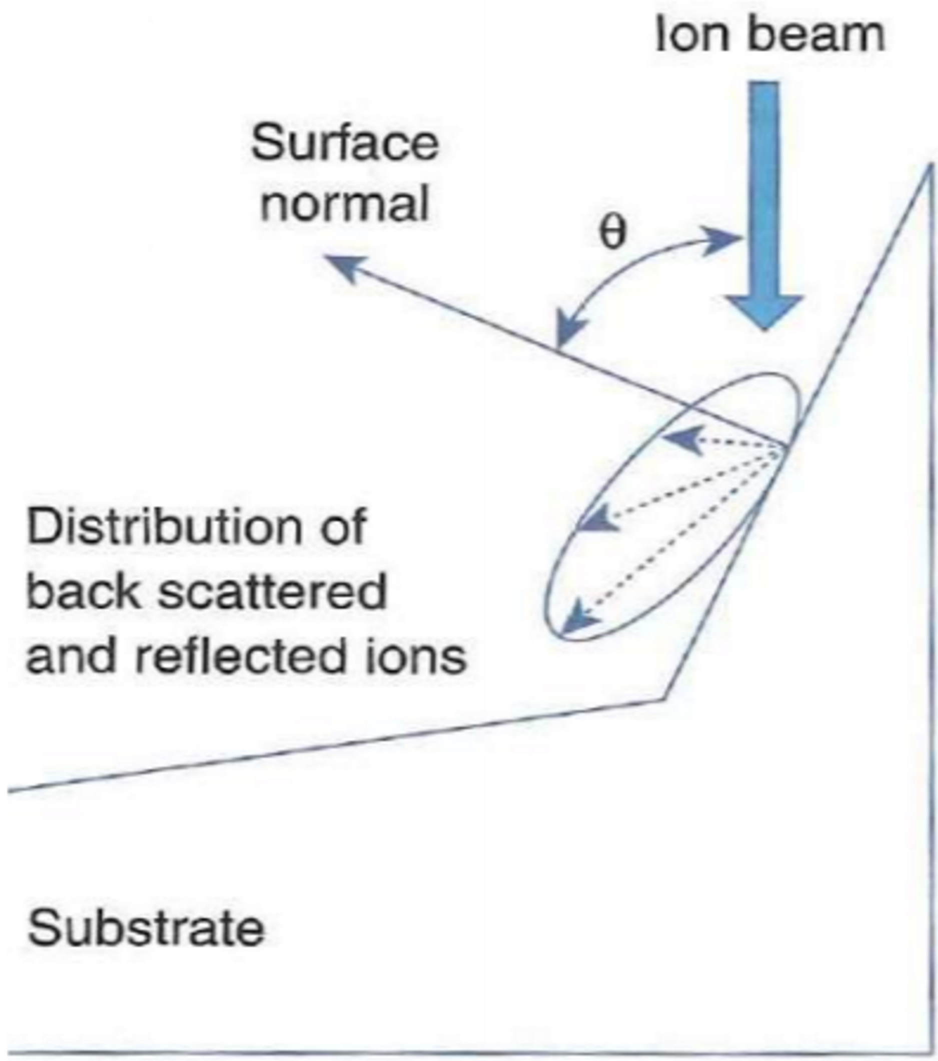




Si atoms are allocated at 2D lattices (for simplification).

A part of ions penetrates into the substrate.

The scattering angle is determined by an impact parameter on incidence.



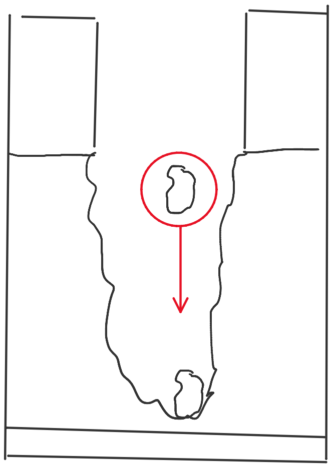
Ion beam gets deflected by tilted faces *

Feature Model 2D - Drop

After the mesh is updated, a drop algorithm is called to fix the floating cells/clusters.

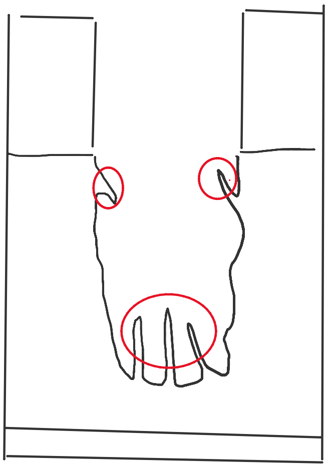
Drop

Sunday, September 13, 2020 6:50 PM



When a cluster is detached/separated from the main feature, the cluster will drop to bottom and reconnect/deposit on surface.

Search for separate shape in topology is a classical problem.



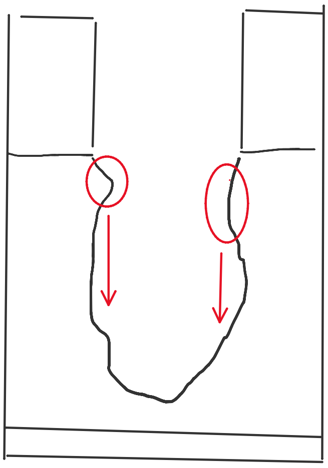
Ocassionally, some branching shapes occur due to the statistical phenomenon. Mechanically, they could fall due to gravity.

Supporting/connecting force

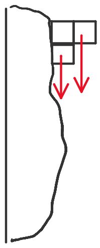
vs.

Gravity

This is very difficult for practical implementation.



Assumption: No horizontal connection.



as long as there is no supporting directly underneath the cell, the cell drops to bottom.

The resulting profine is exactly CONCAVE.

Dropping materials are specified in yuml file. e.g. "S_i" mat is dropping, but "PR" is not.