Documentation of: Scripting for preliminary studies on

**Micro Patterning and Roughness** (MPR)

This initial documentation refers to the version created for APS 2016 (currently, Master version in the repository). Other branches are being developed to look at different problems (motivation, surface geometry…) that can be studied using the same approach.

A. Lasa, Feb 2017

1. **INTRODUCTION:**
2. **MOTIVATION:** 
   * Plasma-material interactions erode plasma facing surfaces, limiting component lifetime and producing impurities that contaminate the plasma
   * Impurity production (sputtering and reflection yields) depends, among other parameters on the impact angle, determined both by local plasma conditions and surface morphology.
   * For rough surfaces *O*(~μm), the local impact angle of a particle differs from the global one *(Fig.1)*.

B

~µm

ion

nglobal

^

nlocal

^

αlocal

αglobal

Fig. 1: a sketch of the concept of global and local angles for particles impacting on a rough surface

* + So far, studies targeted at bridging these parameters have been of interpretative focus, studying the evolution of naturally occurring surfaces exposed to plasma [1]. Significant development has also gone into models treating roughness at atomic scales through fractal surfaces. [2]
  + Here we are interested in larger scale lengths (~μm-mm), bridging the gap between the atomic scale data for sputtering yields and the measurable macro-scale erosion.

1. The **GOAL** of our project is to answer:

* Can we **tailor surfaces** to take advantage of the angular dependence, and **control impurity production and migration?**
* Under which conditions will roughness self-enhance or -smoothen?
* What is the effective dependence of local erosion and migration on global (control) parameters?
* how can this information be integrated in a global modeling framework?

1. **WE PRESENT** the development of an approach to the impact of surface geometry on erosion and re-deposition

* deriving local impact angle and density distributions for rough surfaces, needed to derive spatially resolved erosion and re-deposition
* running sensitivity scans over the main parameters to test different scenarios and reduce uncertainties

1. Ultimately, experimental and theoretical approaches will be combined towards achieving predictive capabilities, including the development of intentionally sculpted surfaces to control impurity production and migration.
2. **ALGORITHM**

To systematically study the relationship between local and global angles the following algorithm has been developed:

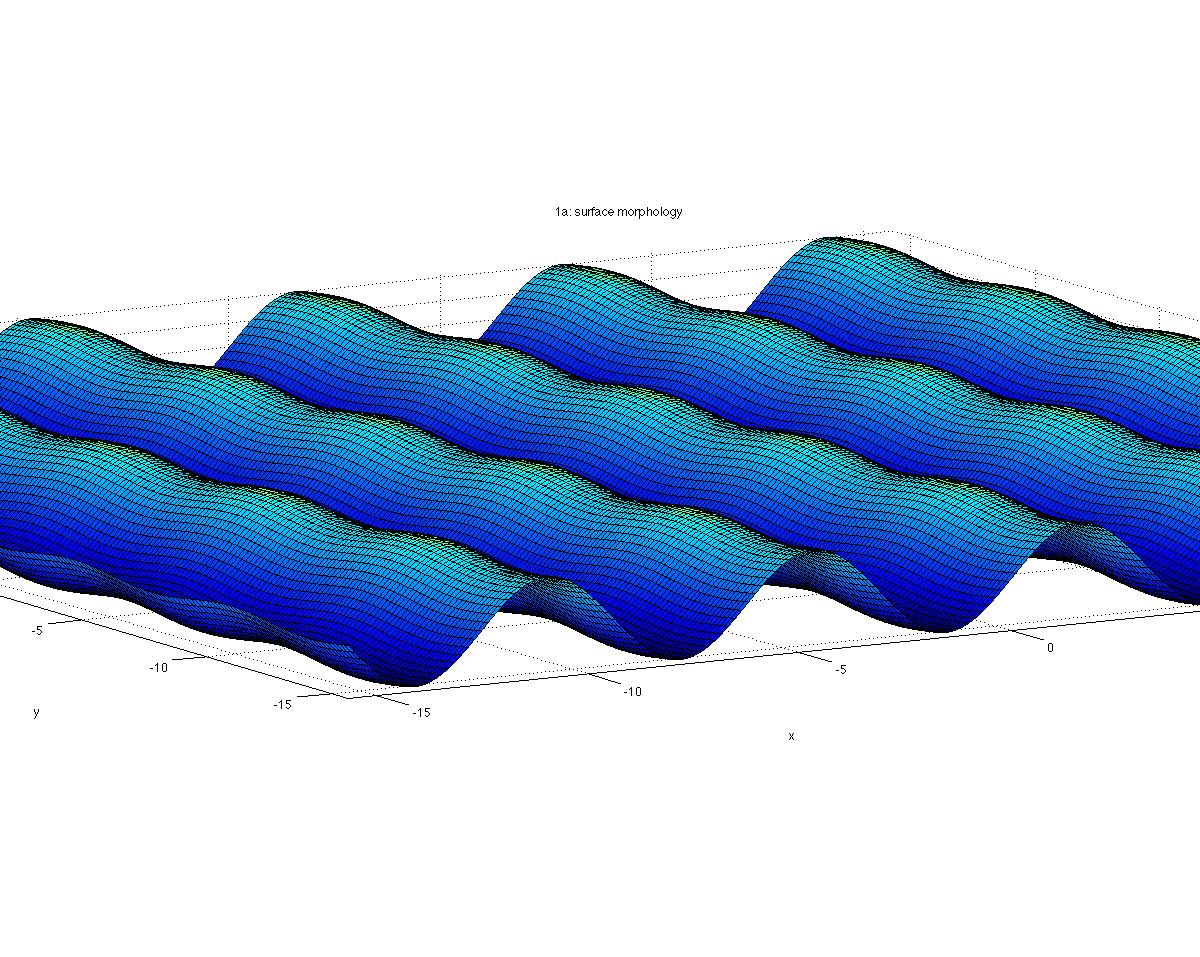


Fig. 2: an incoming particle’s set-up, highlighting the main geometrical parameters

**ns**

φ

θin

Z

X

Y

**(x0, y0)**

**(xs, ys)**

αloc

* 1. **Set up** (Fig. 2):



(Eq. 1)

* a set of particles *O(~106)* are launched
* from random initial positions (x0, y0),
* with straight trajectories (Eq 1):
* towards an analytically described surface (Eq. 2):



(Eq. 2)

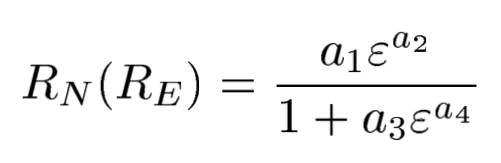
Note: by approximating the trajectory to a straight line we assume that the gyro-radius (usually, *O(mm*)) is much larger than the dimensions modeled here (*O(μm)*).

1. **Analysis of the impact** (point of intersection between zs and zp)

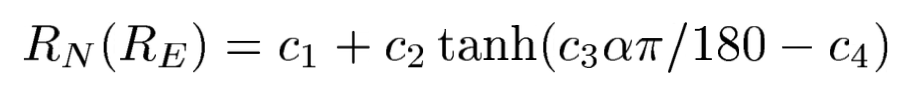
* Main outcome: local impact density (flux) and angle
* Cumulative distributions and profiles (along x and y) of the flux and angle

1. **Reflection and sputtering yields**

Using the local angle and flux, and the Eckstein fit formula, the sputtering and reflection yields are calculated for each cell

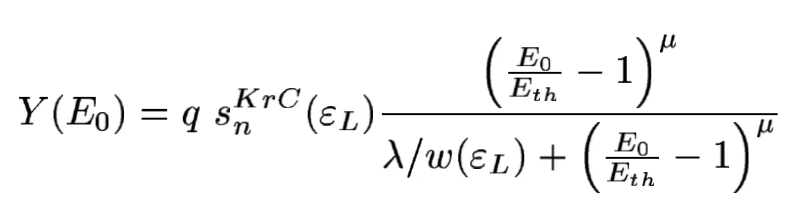


Energy dependence of particle/energy reflection

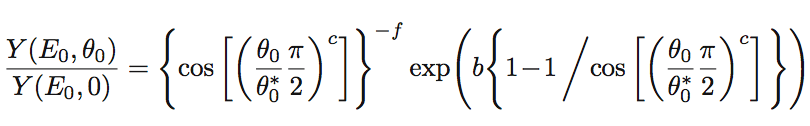


Angular dependence of particle/energy reflection

Energy dependence of the sputtering yield



Angular dependence of the sputtering yield



(Eq. 3 – 6)

where *εL* is the reduced energy; *Eth* the threshold energy; *{q,λ,μ}* and *{ai}* are parameters that depend only on the target-projectile combination; *{f,b,c}* and *{ci}* also depend on impact energy. *Eq. 3-4* are taken from [3] with corrections from [4]; *Eq 5-6* are from [5], with a change of sign in c4 to avoid divergence

[3] R. Behrisch and W. Eckstein, Sputtering by Particle Bombardment, Chapter “The Sputtering Yield”

[4] ftp://ftp.rzg.mpg.de/pub/ipp/eckstein/rep05/errors\_chapter4.pdf

[5] W. Eckstein, Reflection, IPP 17/12 August, 2009

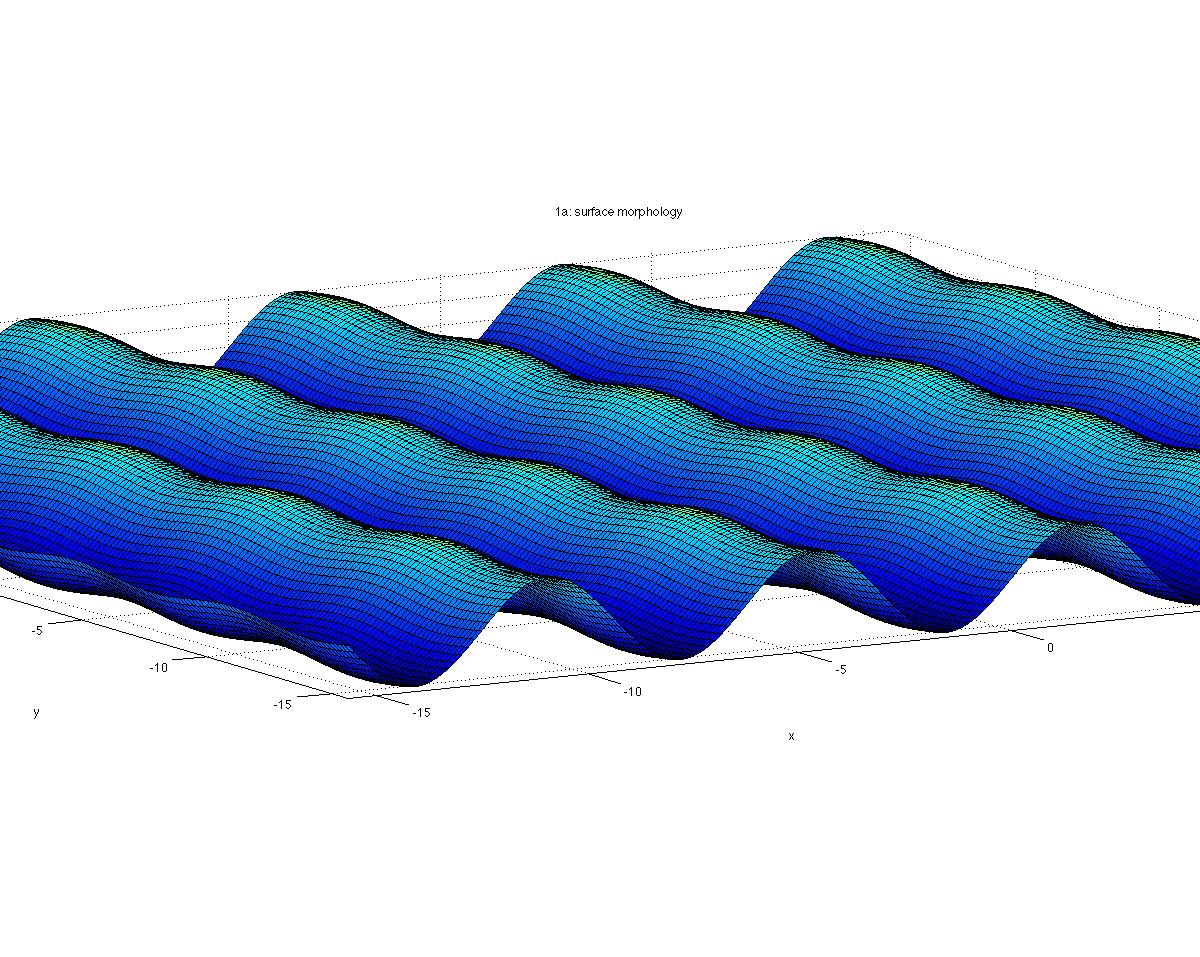


Fig. 3: the set-up of an emitted particle, highlighting, highlighting the main geometrical parameters

y

x

(xs, ys)

(xredep, yredep)

ns

(θout, φout)

1. **Emission of sputtered and reflected particles (Fig. 3)**

* From the point of sputtering/reflection on the surface, as two new populations



(Eq. 7)

* Particles emitted due to sputtering:
  + cosine like distribution in θout:
  + a uniform random distribution in φout
* Particles emitted by reflection:
* cosine-like angular distribution in θout (as for sputtered particles)
* Specular reflection, wrt the local surface normal, in φout

1. **Re-deposition**

Finally, the fraction and location of re-deposited particles is calculated for each population, assuming straight trajectories, zp(θout, φout)

Note:

* by approximating the outgoing trajectory to a straight line, we assume that the ionization mfp is larger than the surface roughness (z-dimension) modeled here.
* as all emitted particles which hit the surface are considered re-deposited, low impact (and thus outgoing) energy is assumed.

1. **PHYSICS**

For the cases tested, output, discussion and conclusion (i.e., physics) see the poster presented by A. Lasa et al., APS 2016.

1. **THE MATLAB SCRIPT**

All input variables are specified in the main file (main.m). Other parameters, such as surface geometry or Eckstein fit parameters for sputtering and reflection yields, are specified in additional input files (e.g., Eckstein folder). The main file calls all other modules needed to perform the **algorithm** described in Sect. 2, in the following order:

* main

define input parameters and call the following modules

1. zs\_zp\_intersect

Calculate intersection of surface and trajectory, and local impact angle (in function part\_surf\_local\_angle). Including the following functions:

* + part\_surf\_local\_angle: given an analytically described surface and particle trajectory, finds the intersection point and angle wrt the surface normal

1. impact\_output\_and\_distributions

Calculate distributions; get matrixes ready to output and plot. Including calls to the following functions:

* + surfacegrid: create a gridded version of the surface
  + histogram\_and\_distr: find the distribution of number of impacts
  + angle\_profile\_ave\_and\_distr: find the profile, average and distribution of the local impact angle

1. cellarea

calculate area of each grid cell, as in a rough surface, a uniform grid formed as (xmax-xmin) / npoints will lead to different areas

1. localflux

scale to obtain flux reaching each area; local\_flux = Ncounts / cell\_area;

1. plot\_incoming\_impacts

plotting incoming particles output: surface morphology, impact angle and flux

1. erosion

Erosion of the target by the projectiles for the local conditions (flux and angles) calculated above, and using the Eckstein formula

1. reflection\_RN

Projectile (particle) reflection for the local conditions (flux and angles) calculated above, and using the Eckstein formula

1. reflection\_RE

Projectile energy reflection for the local conditions (flux and angles) calculated above, and using the Eckstein formula

1. particle\_emission

particles reflected (species=projectile, E = energy reflection coeff. & angle = near specular reflection) and sputtered (species=target, E = Thompson distribution & angle = under/over-cosine) are emitted from the surface in straight lines. the function that finds the possible intersection of emitted particles with the surface. It also calls the following functions:

* thompson\_distr: creates a Thompson distribution, used for the (outgoing) energy of sputtered particles)
* cosn\_distr: creates a under / over cosine distribution, used for the (outgoing) angle of sputtered particles)
* emitted\_part\_surf\_intersec: given an analytical surface and particle trajectory, find the intersection point between the emitted particle (by sputtering or reflection) and the surface, if it exists. If the particle impacts on the surface, it is considered re-deposited.

1. plot\_sputtering\_and\_reflection

plotting of reflection and sputtering output

Further, the **surface and particle trajectory** are described in:

1. Surface description functions:

* surface function: zs
* Surface derivative: dzs
* Surface derivative wrt x: dzsx
* Surface derivative wrt y: dzsy

1. Particle trajectory:

* trajectory wrt x-coord : zpx
* trajectory wrt y-coord : zpy

A detailed description of the algorithm, with a cleaned-up version of the script and well commented, is given in ‘matlab\_script\_description’.

1. ANGULAR DISTRIBUTION OF IMPACTS

We can now use a distribution in impacts (instead of mono-angular theta).

There’re multiple distributions already implemented:

From Borodkina’s work: I. Borodkina et al., Contrib. Plasma Phys.56, No. 6-8, 640 – 645 (2016)

* There are 3 distributions: for 85, 88 and 89 degrees.

From Curreli’s work: Physics of Plasmas 22, 043503 (2015)

* We implemented one distribution, for 85deg.

From Chrobak’s work: Nucl. Fusion 58 (2018) 106019

* We’ve implemented the case for He plasma ‘Case R-He’

The process is:

1. Get the data: either from authors, or using ’grabit’ app of matlab, which saves the data into a matrix in matlab format
2. Fit to a n-gaussian: run the curve-fit tool (cftool) ; choose the 1st column of the fit at x, 2nd column as y, change fit to ‘gaussian’ and choose the degree (~2-5, depending on the number of peaks)
   1. If the matlab data is as single variable, open the workspace, select and copy the 1st column and do A=(paste); idem for B=2nd column
   2. Then, if cftool, used x=A and y=B
3. Save the fit (if you want) and copy the parameters (a1, b1, c1, a2,…) to your case (as ga1, gb1, bc1, ga2…) in ‘zs\_zp\_intersect.m’ file – use the other cases as reference. Modify the maximum of the distribution (currently 0.08—0.09) and range (currently 45-90deg) as needed.