

```
# -*- coding:utf-8 -*-  
#  
# written by Shotaro Fujimoto  
# 2016-08-19
```

Modules

[numpy](#) [matplotlib.pyplot](#) [time](#)

Classes

[growing_string.Main](#)([base.Main](#))

[Roughness](#)

class **Roughness**([growing_string.Main](#))

Method resolution order:

[Roughness](#)
[growing_string.Main](#)
[base.Main](#)

Methods defined here:

__init__(self, L=60, frames=1000, beta=0.0)

Methods inherited from [growing_string.Main](#):

calc_weight(self, s, i, r_i=None, r_rev=None)
ベクトルの内積を元に、Boltzmann分布に従って成長点選択の重みを決定

cleanup_bonding_pairs(self, key, index_start, index_stop)

dot(self, v, w)
0〜5で表された6つのベクトルの内積を計算する。

v, w (int): ベクトル(0〜5の整数で表す)

get_bonding_pairs(self, s, index_start, index_stop)

get_neighbor_xy(self, key)
Stringクラスのインスタンスsの隣接する非占有格子点の座標を取得する

s (String): 対象とするStringクラスのインスタンス

plot_all(self)
軸の設定、三角格子の描画、線分描画要素の用意などを行う

ここからFuncAnimationを使ってアニメーション表示を行うようにする

plot_string(self)
self.strings内に格納されているStringを参照し、グラフ上に図示する

start_animation(self, filename='')

update(self, num=0)
FuncAnimationから各フレームごとに呼び出される関数

1時間ステップの間に行う計算はすべてここに含まれる。

update_each_string(self, key)

Methods inherited from [base.Main](#):

create_random_strings(self, N=3, size=[10, 5, 3])
Create N strings of each size is specified with 'size'.

This process is equivalent to self-avoiding walk on triangular lattice.

get_next_xy(self, x, y, *vec)

Functions

```
eval_fluctuation_on_surface(self, pos, test=False)

eval_std_various_width(theta, r, R_t)

fitting_manual(fig, ax, x, y)

plot_result(beta, frames, x, y, ax=None)

plot_to_verify(beta, frames, theta, r, R_t, label_list, save_image=False)

start(beta, frames, plot_to_verify_bool=True, save_image_bool=False)
```