

points

```
# -*- coding:utf-8 -*-
#
# written by Shotaro Fujimoto
# 2016-04-22
```

Modules

[numpy](#)

Classes

[Points](#)class **Points**

Methods defined here:

__init__(self, N, position_x, position_y, natural_length, K, length_limit)

Initialize class variants.

--- Arguments ---

N (int) : How many points should placed
 position_x (ndarray): Array of the valuse of x axis for each points
 position_y (ndarray): Array of the valuse of y axis for each points
 natural_length (ndarray): Array of natural length of each strings
 K (ndarray): Array of spring constant
 length_limit (float) : Threshold for dividing to 2 strings

create_new_point(self, k, X)

新しい点を2点の間に追加し、各物理量を再設定する

k番目とk+1番目の間に新しい点を追加

divide_if_extended(self, X)

2点間の自然長がlength_limitの設定値より大きいとき新しい点を追加する

get_distances(self, x, y)

Caluculate distance between two points and return list.

--- Arguments ---

x (list or ndarray): x座標の値のリスト
 y (list or ndarray): y座標の値のリスト

grow(self, func_nl, func_k)

2点間の自然長を大きくする & バネ定数を変化させる

バネ定数は単位(自然長)長さあたりが同じいなるように随時変化させる

--- Arguments ---

func_nl (function): N-1(開曲線), N(閉曲線)次元のnp.arrayに対する関数
 返り値は同次元のnp.arrayで返し、これが成長後の自然長のリストである
 func_k (function): N-1(開曲線), N(閉曲線)次元のnp.arrayに対する関数
 返り値は同次元のnp.arrayで返し、これが成長後のバネ定数のリスト

update_natural_length(self, k, d)

自然長を更新

Called from self.**create_new_point**
 Change: self.**natural_length**

update_point_position(self, k)

点を追加

Called from self.**create_new_point**
 Change: self.**position_x**, self.**position_y**

update_point_velocity(self, k)

速度を更新

Called from self.**create_new_point**
 Change: self.**vel_x**, self.**vel_y**

update_spring_constant(self, k)

バネ定数を更新

Called from self.**create_new_point**
 Change: self.**K**