Surface Growth with Random Tetris Pieces

Le Chen, Mauricio Montes, Ian Ruau

Dec 02, 2023

CONTENTS

CHAPTER

ONE

RD_CLI MODULE

RD_CLI.**Ballistic_Deposition**(*width*, *height*, *steps*)

Simulate Ballistic Deposition on a substrate. In this simulation, particles stick upon contact with the substrate or a deposited particle.

This is simulation for snowflakes piling.

Parameters

- **width** (*int*) – Width of the substrate.
- **height** (*int*) – Height of the substrate.
- **steps** (*int*) – Number of particles to drop.

Returns Filename of the output substrate for Ballistic Deposition; A csv file contains the substrate state.

Return type string

RD_CLI.**Envelop**(*substrate*)

Compute the top envelope of a substrate.

This function calculates the highest particle position in each column of the substrate. It is used to visualize the top envelope of the substrate in the simulation.

Parameters **substrate** (*numpy.ndarray*) – The substrate matrix to compute the envelop for.

Returns Array representing the top envelope of the substrate.

Return type numpy.ndarray

RD_CLI.**Random_Deposition**(*width*, *height*, *steps*)

This is a function to simulate Random Deposition on a substrate. This is simulation for the independent boxes (sand) piling. All columns are independent.

Parameters

- **width** (*int*) – Width of the substrate.
- **height** (*int*) – Height of the substrate.
- **steps** (*int*) – Steps or times to run.

Returns Filename of the output substrate.

Return type string

RD_CLI.**Random_Deposition_Surface_Relaxation**(*width*, *height*, *steps*)

This is a function to simulate Random Deposition on a substrate with the surface relaxation. Particle will seek the lowest left/right neighbors to land.

This is simulation for the independent boxes (sand) piling with surface relaxation.

Parameters

- **width** (*int*) – Width of the substrate.
- **height** (*int*) – Height of the substrate.
- **steps** (*int*) – Steps or times to run.

Returns Filename of the output substrate.

Return type string

RD_CLI.**interface_width**(*filename*)

Compute and visualize the interface width of a substrate from a given simulation.

This function reads the substrate state from a file, calculates the interface width over time, and generates a log-log plot of the interface width. It also computes the slope of the log-log plot as a function of time.

Parameters **filename** (*str*) – The name of the file containing the substrate data.

Returns Array containing the interface width calculated at each step; An image file with the same name as the input file, but with a .png extension. The file contains the statistical figures.

Return type numpy.ndarray

RD_CLI.**main**()

The main function to simulate different types of surface growth models based on the provided command-line arguments.

This function sets up a command-line interface for simulating Random Deposition, Random Deposition with Surface Relaxation, or Ballistic Decomposition on a substrate. It accepts various parameters like width, height, and number of steps for the simulation. It also provides options for generating a movie of the simulation and calculating interface width.

The function decides the type of simulation based on the arguments passed, performs the simulation, and then proceeds to calculate the interface width. If the movie generation option is selected, it invokes another script to generate the movie.

To use the script from terminal, the following options are expected:

-w, --width : Width of the substrate (default: 100)

-e, --height : Maximum height of the substrate (default: 60)

-s, --steps : Number of particles to drop (default: 5000)

-r, --relax : Enable surface relaxation (default: False)

-b, --BD : Enable ballistic decomposition (default: False)

-m, --movie : Generate an mp4 movie of the simulation (default: False)

It returns:

1. A text file representing the substrate state.
2. Statistical figures, including a log-log plot for the interface width and the estimated slope.
3. (Optional) An mp4 movie of the simulation process.

Example

```
ptyhon3 RD_CLI.py -w 100 -e 60 -s 5000 --BD --movie
```

In this example, the script will simulate Ballistic Decomposition on a substrate of size 100x60 for 5000 steps. And the simulation movie will be generated.

CHAPTER

TWO

VISUALIZE_RD MODULE

Visualize_RD.**main**()

Main function to execute the particle deposition visualization script.

This script visualizes the process of particle deposition based on the data from a substrate simulation. It generates a video showing the evolution of the deposition over time. The script offers several command-line options to customize the visualization, such as showing the top envelope, the average height, and adjusting the video playback rate.

Options:

-f, --file : Required. Path to the substrate file containing simulation data.

-t, --title : Optional. Title of the plot. If not provided, no title is displayed.

-r, --rate : Optional. Frame rate for the video. Default is 4 frames per second.

-e, --envelop : Optional. Flag to display the top envelope of the deposition. Default is False.

-a, --average : Optional. Flag to display the average height of the deposition. Default is False.

-p, --play : Optional. Flag to automatically play the video after generation. Default is False.

Example:

```
python3 Visualize_RD.py -f path/to/substrate.txt -t "Simulation Title" -r 10
--envelop --average --play
```

This command will run the script on 'substrate.txt', set a title, use a frame rate of 10 fps, and display both the top envelope and average height in the visualization.

Visualize_RD.**visualize_simulation**(*filename*, *plot_title*, *rate*, *envelop*, *show_average*)

Visualize the particle deposition simulation and generate a video.

This function loads the substrate data from a file and visualizes the deposition process as an animation. It supports visualizing the top envelope and average height of the deposited particles. The final output is saved as an mp4 video file.

Parameters

- **filename** (*str*) – The path to the substrate data file.
- **plot_title** (*str*) – The title of the plot.
- **rate** (*int*) – The frame rate for the video.
- **envelop** (*bool*) – Flag to indicate whether to show the top envelope.

- **show_average** (*bool*) – Flag to indicate whether to show the average height.

Returns The filename of the generated mp4 video.

Return type str

CHAPTER

THREE

TETRIS_COMPLETE MODULE

This module simulates the surface growth by Tetris pieces. It includes functions to generate random Tetris pieces, calculate their landing positions on a substrate, and simulate a game of Tetris for a given number of steps and a defined grid size.

By Ian Ruau (iir0001@auburn.edu) and Mauricio Montes (mauricio.montes@auburn.edu) Date: 12/2023

tetris_complete.**Tetris_Choice**()

Randomly selects a Tetris piece and its orientation.

There are 7 Tetris pieces:

- 0 : the square;
- 1 : the line;
- 2 : the L;
- 3 : J;
- 4 : the T;
- 5 : the S;
- 6 : the Z.

There are 4 orientations for each piece:

- 0 is the original orientation;
- 1 is the 90 degree rotation;
- 2 is the 180 degree rotation;
- 3 is the 270 degree rotation.

Returns A 2-element array: the first element is the piece type (0-6); the second element is the orientation (0-3).

Return type numpy.ndarray

To-do:

- add input file to specify the probability of each piece.

tetris_complete.**Tetris_RD**(*width*, *height*, *steps*)

This function simulates the Tetris Decomposition model on a substrate.

Parameters

- **width** (*int*) – The width of the substrate.

- **height** (*int*) – The height of the matrix.
- **steps** (*int*) – The steps to simulate.

Returns Filename of the output file.

Return type string

tetris_complete.**ffnz**(*matrix*, *height*, *column*)

Finds the first non-zero entry in a specified column of a matrix.

Parameters

- **matrix** (*numpy.ndarray*) – The matrix to search.
- **height** (*int*) – The height of the matrix.
- **column** (*int*) – The column index to search in.

Returns The index of the first non-zero entry.

Return type int

tetris_complete.**main**()

To use the script from terminal, the following options are expected:

-w, --width : Width of the substrate (default: 100)

-e, --height : Maximum height of the substrate (default: 60)

-s, --steps : Number of particles to drop (default: 5000)

It returns:

1. A text file representing the substrate state.

Example

```
ptyhon3 tetris_complete.py -w 100 -e 60 -s 5000
```

In this example, the script will simulate Tetris Decomposition on a substrate of size 100x60 for 5000 steps. And the simulation movie will be generated.

CHAPTER

FOUR

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

INDEX