Computational Physics Homework Report

Fateme RamezanZade-99100693 Problem Set 3

1. **Random Deposition with Relaxation**

1.1 Functions Description

Mean:

This function takes an array, calculates sum of all its elements and divides the sum by the number of elements. Then returns the amount, which is the mean.

Dev:

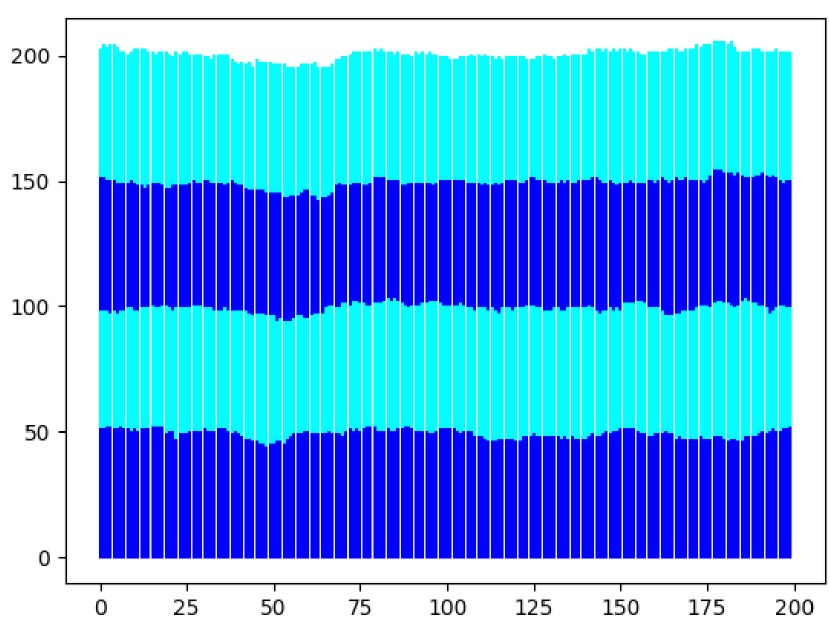
This function takes an array, first calculates the mean of its elements and then calculates the mean if the squared amount of its elements. Finally, uses standard deviation formula for them and returns the result which is the standard deviation of the elements of array.

1.1 Main Code

Firstly, an array of 200 elements is defined and each element is set equal to zero. Then in a loop which repeats 10000 times, a random number is chosen, then for this number and the previous and the next one the elements of the array are checked. (This value corresponds to the height of that column so I call it height.) Then the minimum height is increased by one. If they are all equal, the height if ith column is increased, and if the height of the right side and left side are equal and smaller than the ith column, one of them is randomly chosen. It is worth mentioning that the boundary condition is chosen to be periodic, so the first and last element are treated as neighbors. After the loop ended, a copy of the resulted array is made. This process is repeated three times. Now for each element of the array, we plot some vertical lines. The first is plotted from the x axis up to the first copy of h, with blue color. Then we plot a line from the first copy to the second one, with cyan color. We repeat this two other last time with blue and cyan. This process of changing lines helps us see how the upper edge would look at some definite times.

1.2 Results

Each time before making a copy, the mean amount of h and its standard deviation is calculated.



$$\bar{h}(10000\tau) = 50$$

$$w(10000\tau) = 1.8055$$

$$\bar{h}(2 \times 10000\tau) = 100$$

$$w(2 \times 10000\tau) = 1.7944$$

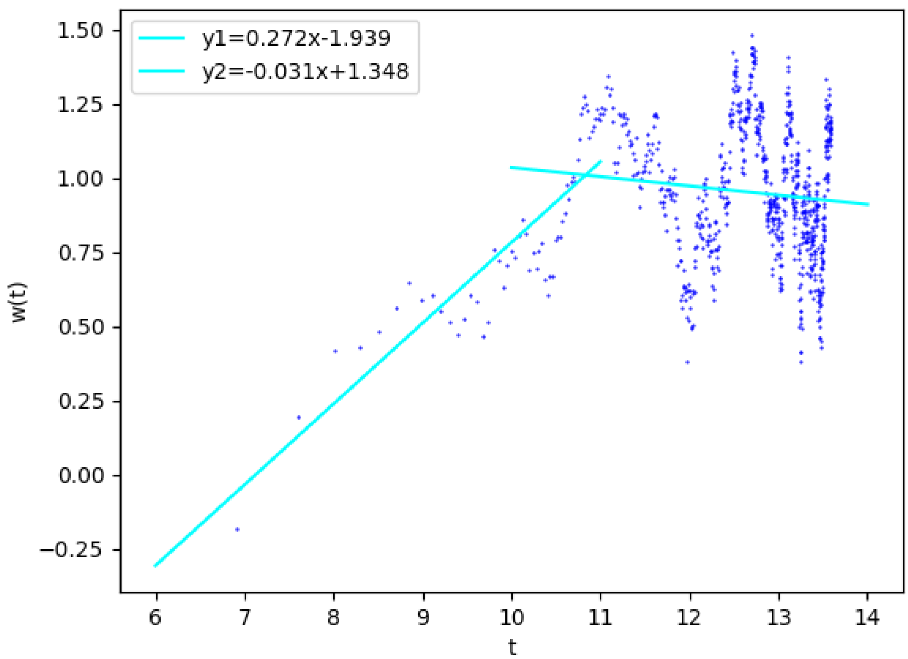
$$\bar{h}(3 \times 10000\tau) = 150$$

$$w(3 \times 10000\tau) = 2.1236$$

$$\bar{h}(4 \times 10000\tau) = 200$$

$$w(4 \times 10000\tau) = 2.2360$$

We can see that $\bar{h} = \frac{t\tau}{l}$ as we expected. If we want to achieve a shape with smoother edges, we need more time. in the following image, each row is plotted after 100000τ. Now we want to find out how w changes with time. Here I have plotted a log-log graph of mean of w(t) for 10 ensembles for each 10000τ:

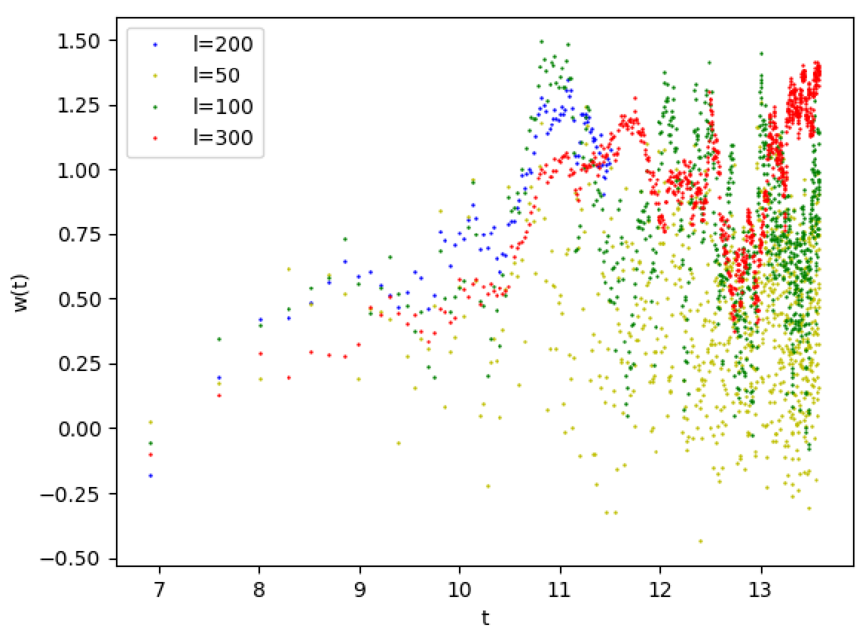


The lines are fitted to each segment using least squares method. We find the intercept of these two:

$$0.272x - 1.939 = -0.031x + 1.348 \Rightarrow \log{(t_s)} = 10.848\,, \log{(w_s)} = 1.011$$

It is worth mentioning that increasing the number of ensembles would give a better graph. Also, saturation could be seen more clearly if we plotted data for a longer time. from this plot, $\beta = 0.272$

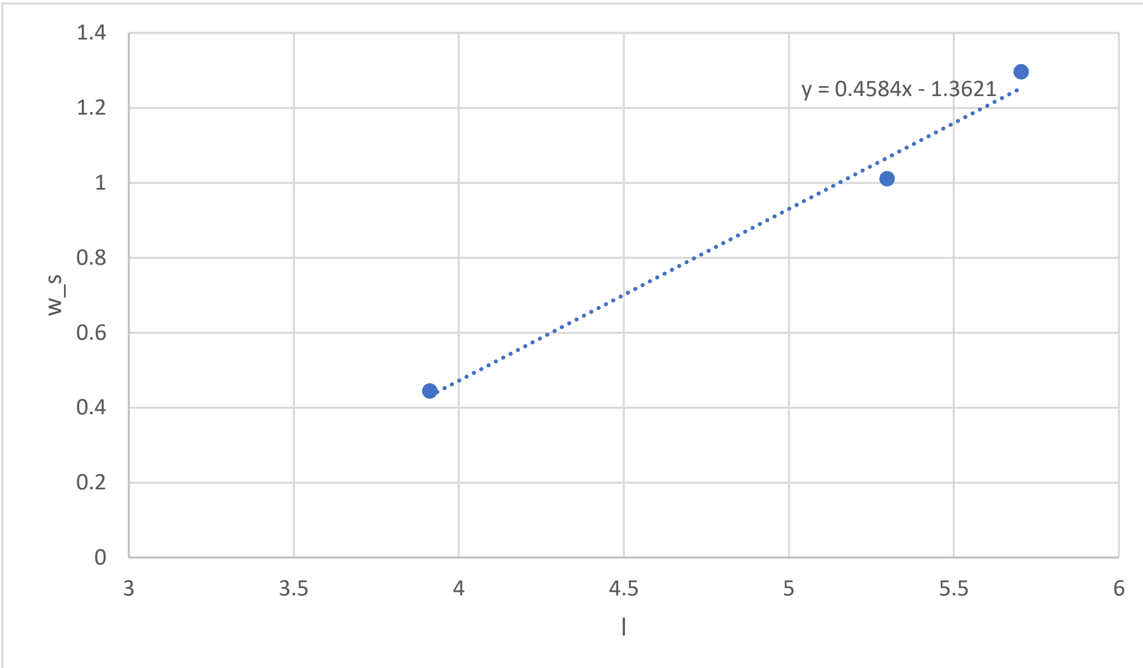
Now we plot the previous graph for different base lengths:



Here is the table of $\log(w_s)$ for different base lengths (this value is calculated using the same method we did for 200):

l	50	200	300
$\log(w_s)$	0.444	1.011	1.296

Now we can find α using the log-log graph of $w_s - l$:



Thus we can conclude that $\alpha = 0.458$

2. **Ballistic Deposition**

2.1 Functions Description

Height:

This function takes an array and searches for the last element in it which its value is 1. Then returns the number if that element.

Dev:

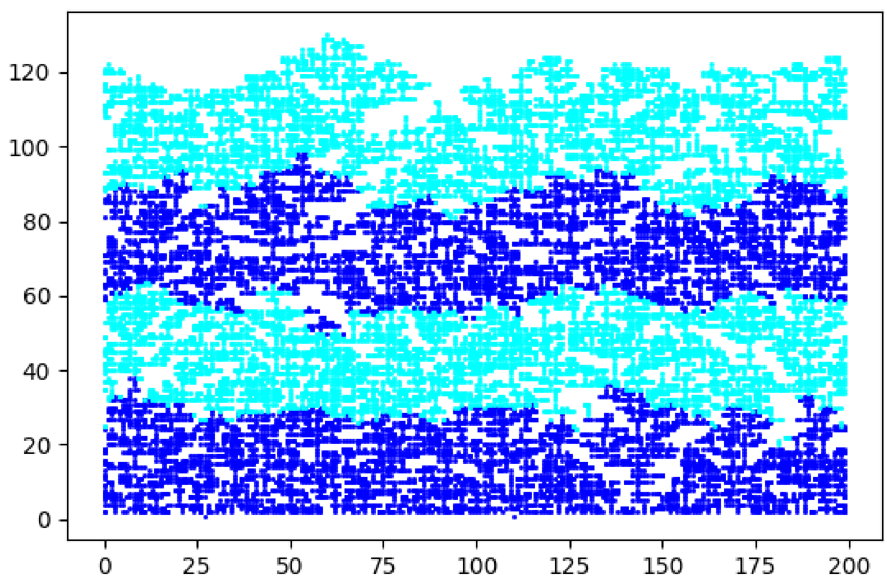
This function takes an array, and uses np.var function to calculate the variance of its element. Then returns the square root of it which is the standard deviation.

2.2 Main code

Firstly, an array of “l” elements is defined and each element is set equal to zero. Then in a loop which repeats “t” times, a random number “i" is chosen, then for this number and the previous and the next one the elements of the array are checked. The height of each is checked using the Height function, and the h[i][x] element of array h is set equal to 1, where x is the bigger height of neighbor elements. Then this point is plotted with a color based on the number of t. It is worth mentioning that the boundary condition is chosen to be periodic, so the first and last element are treated as neighbors. At the end of the loop, each h[i] is appended by one zero element. If the number of the turn is a multiple of "t/4", the mean height and standard deviation is calculated.

2.3 Results

This figure is plotted with l=200 and t=20000



$$\bar{h}\left(\frac{t}{4}\tau\right) = 28.4$$

$$w\left(\frac{t}{4}\tau\right) = 2.8175$$

$$\bar{h}\left(\frac{t}{2}\tau\right) = 58.31$$

$$w\left(\frac{t}{2}\tau\right) = 3.0704$$

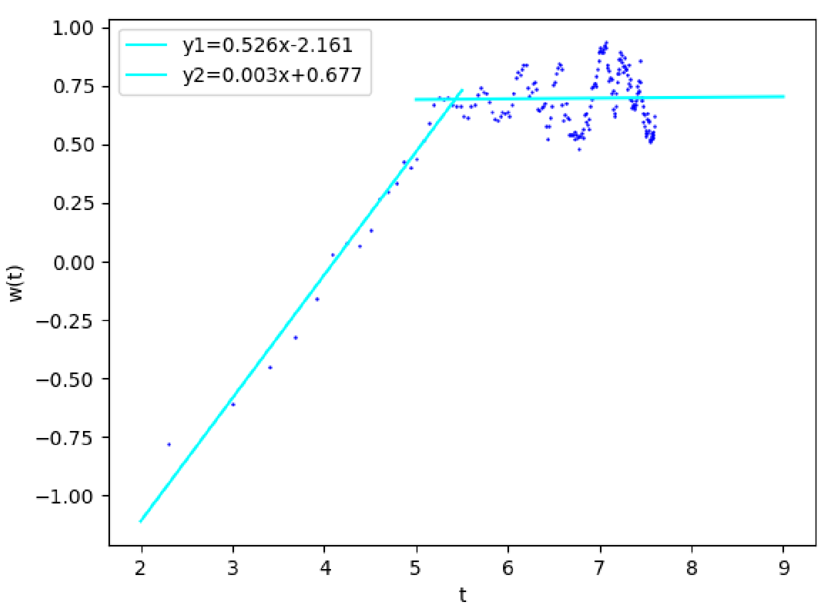
$$\bar{h}\left(\frac{3t}{4}\tau\right) = 88.1$$

$$w\left(\frac{3t}{4}\tau\right) = 3.8832$$

$$\bar{h}(t\tau) = 118.03$$

$$w(t\tau) = 3.2138$$

Now we want to find out how w changes with time. Here I have plotted a log-log graph of mean of w(t) for 10 ensembles where l=50 and t=2000 for each 10τ:

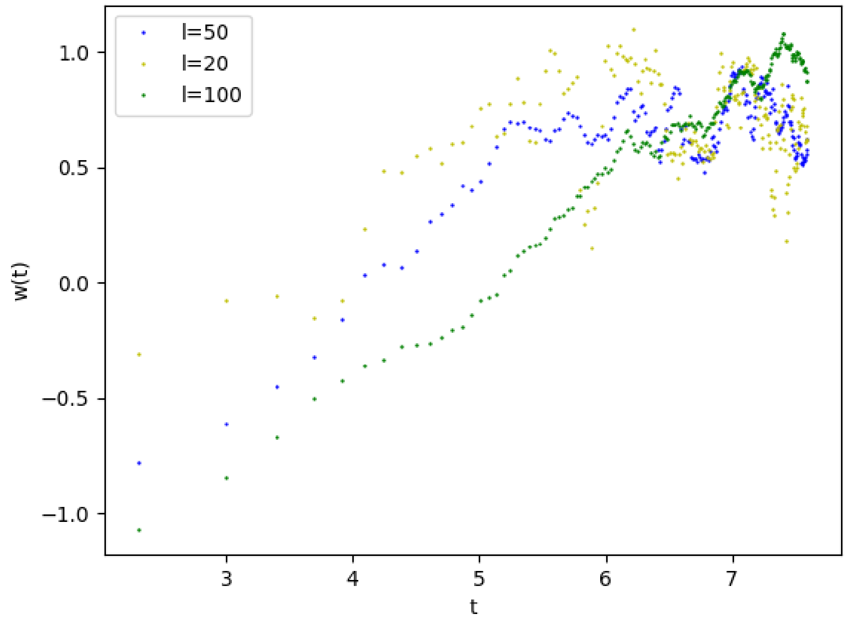


The lines are fitted to each segment using least squares method. We find the intercept of these two:

$$0.526x - 2.161 = 0.003x + 0.677 \Rightarrow \log(t_s) = 5.42\, , \log(w_s) = 0.693$$

from this plot, $\beta = 0.526$

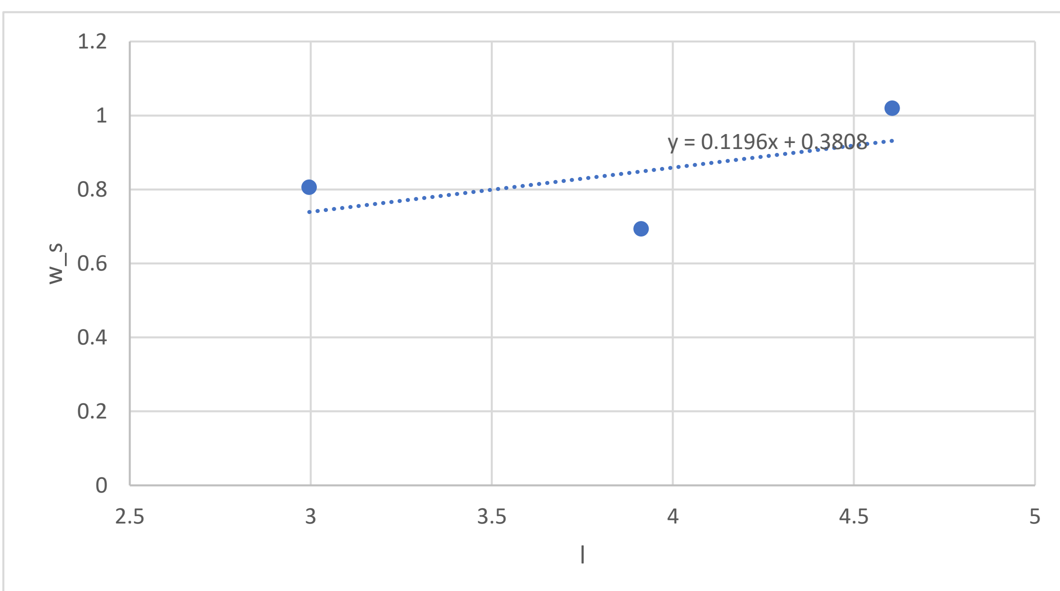
Now we plot the previous graph for different base lengths:



Here is the table of $\log(w_s)$ for different base lengths (this value is calculated using the same method we did for 200):

l	20	50	100
$\log(w_s)$	0.806	0.693	1.020

Now we can find α using the log-log graph of $w_s - l$:



Thus we can conclude that $\alpha = 0.119$

3. Ballistic Deposition (from a point)

3.1 Functions Description

Height:

This function takes an array and searches for the last element in it which its value is 1. Then returns the number if that element.

Dev:

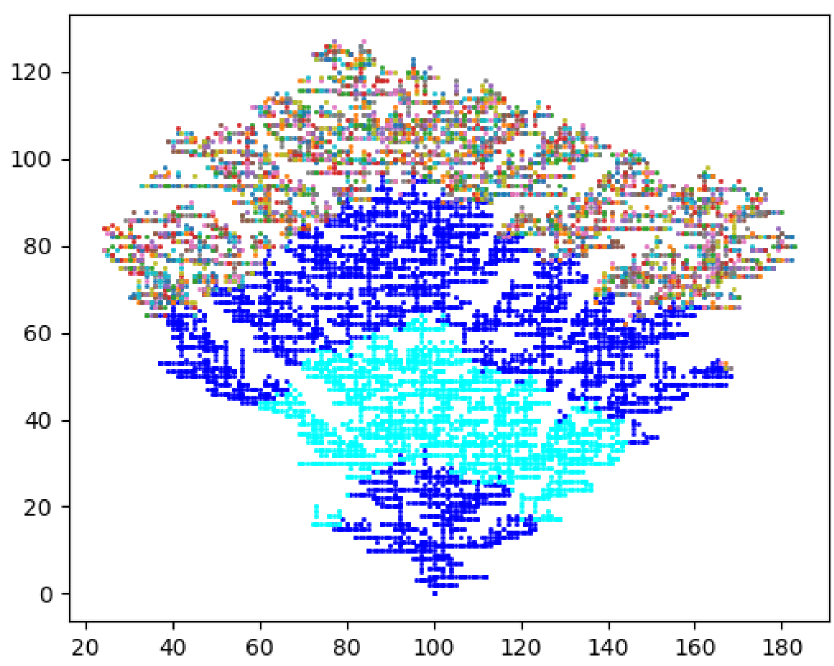
This function takes an array, and uses np.var function to calculate the variance of its element. Then returns the square root of it which is the standard deviation.

3.2 Main code

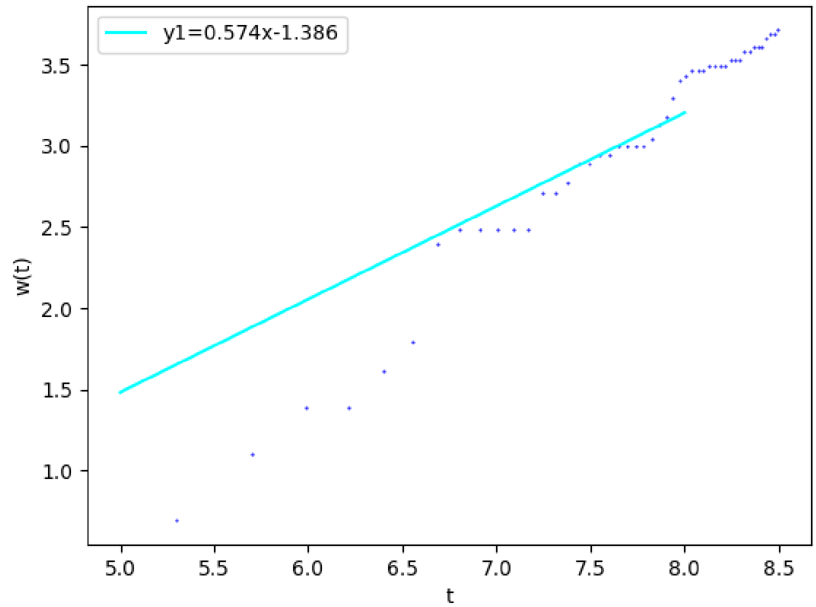
Firstly, an array of "l" elements is defined and each element except the one in the middle is set equal to zero. The middle one is set equal to 1. Then in a loop which repeats "t" times, a random number "i" is chosen, then for this number and the previous and the next one the elements of the array are checked. The height of each is checked using the Height function, and the h[i][x] element of array h is set equal to 1, where x is the bigger height of neighbor elements. Then this point is plotted with a color based on the number of t. It is worth mentioning that the boundary condition is chosen to be periodic, so the first and last element are treated as neighbors. At the end of the loop, each h[i] is appended by one zero element. After that, on order to find the width of the shape every 400τ, we check every row of h and find the place of the first and the last element which is equal to one. Then we take the maximum length to be the width of our shape.

3.3 Results

Result for t=20000



Here is the log-log plot if width over time:



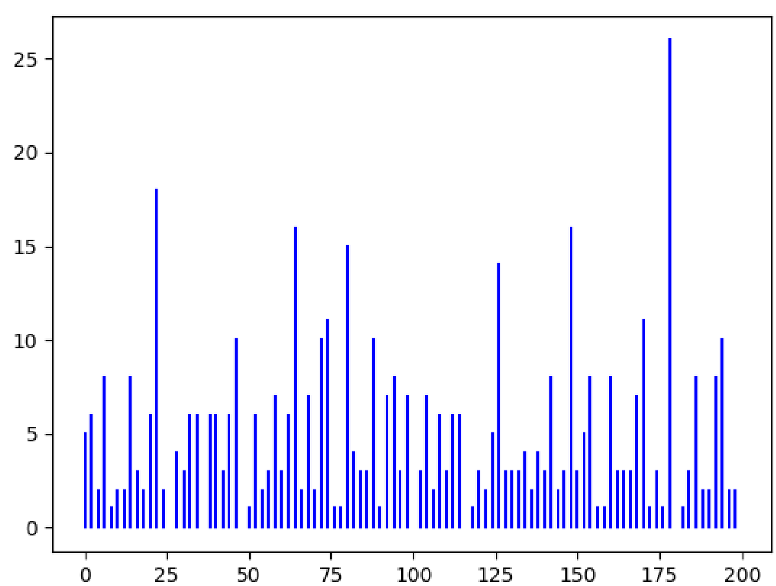
4. Compeatative Growth

4.1 Main Code

Firstly, an array of 200 elements is defined and each element is set equal to zero. Then in a loop which repeats 8000 times, a random number is chosen. Then the elements before it are checked to see if their height is equal to height and the element with that number from the array is increased by 1. Now for each element of the array, we plot some vertical lines which are plotted from the x axis up to h.

4.2 Results

l=200 , t=1000, angle=60degrees



l=200 , t=5000, angle=60degrees