growing snowflakes

using diffusion limited aggregation

By Enoch Lau

Bibliography

- http://www.cpan.org/modules/by-module/Math/Math-Fractal-DLA-0.20.readme
- “Fractal Geometry” http://classes.yale.edu/fractals/Panorama/Physics/DLA/DLA.html
- “What is a snowflake?” http://www.yptenc.org.uk/docs/factsheets/env_facts/snow.html

Overview

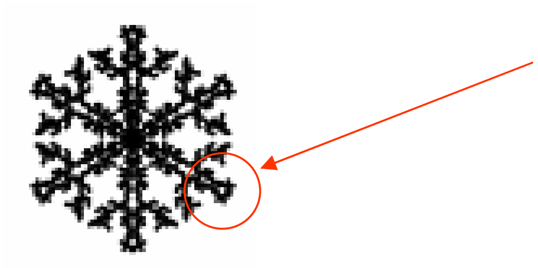
- Discussion of what DLA is
- Issues in designing a DLA algorithm
- Empirical analysis of output from computer program
- Analysis of dimension of DLA fractals

Introduction to DLA

- *Diffusion Limited Aggregation* (DLA) is where you create clusters of particles by moving them randomly towards a target area until they collide with the existing structures and then become part of the cluster.
- It was invented by two physicists T.A. Witten and L.M. Sander in 1981.

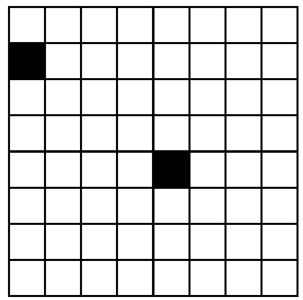
Introduction to DLA

- They are classed into a family of fractals known as *stochastic fractals* because of their formation by random processes.
- As such, they do not possess self-similarity, although they may exhibit similar types of structures at different scales.

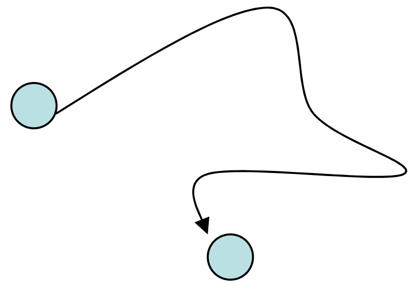


Types of DLA

- There are two ways to simulate DLA on computer:
  - Grid based – the particles are represented by on and off states in a grid-like system, and are constrained to move in integral steps



  - Particle based (Brownian motion) – the particles are able to move freely, and are allowed to join the main cluster at any angle



My Implementation of DLA

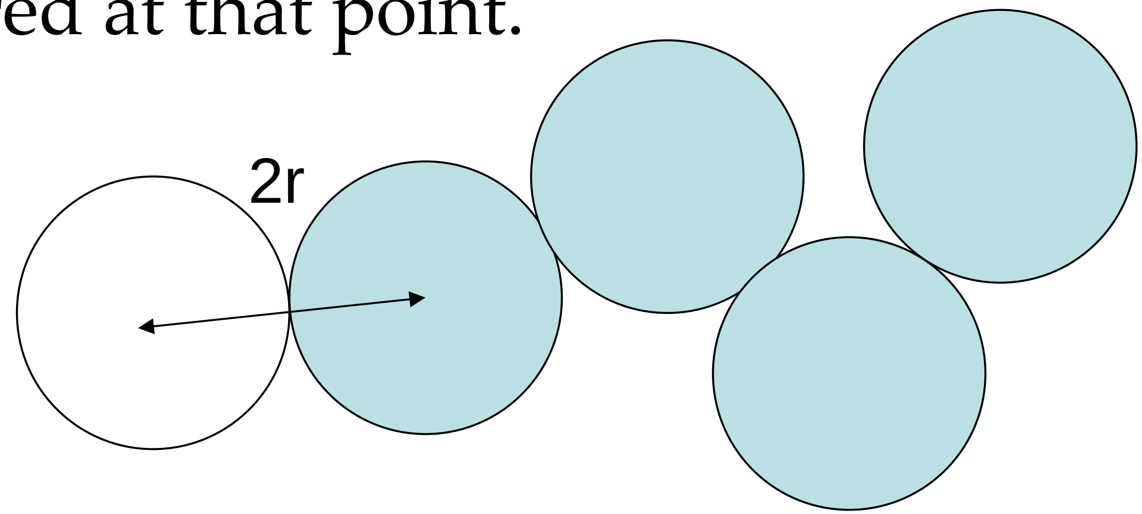
- My implementation of DLA is written in VB.NET using a particle-based system.
- This involves an object-oriented programming approach
- You start off with a single particle in the centre of the output screen, and new particles are produced from one of the corners.

My Implementation of DLA

- For each step ("tick"), the particles change their position using these two motions:
  - A random motion – a transform is chosen from a set of fixed transforms (in this case, two dimensional matrices), which were generated at random. The fixed set of transforms speeds it up somewhat, and also allows for more evident structures to form than possible if it were purely random.
  - An attractive motion – the particle is attracted towards the centre of the screen, otherwise the rendering time would be prohibitively long.

My Implementation of DLA

- The particles are assumed to have a radius, and when the freely moving particle moves within this radius of a particle forming part of the existing cluster, the freely moving particle is captured at that point.



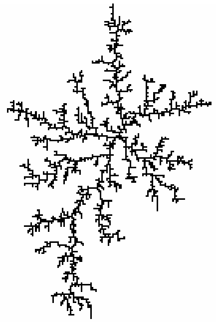
Algorithmic Considerations

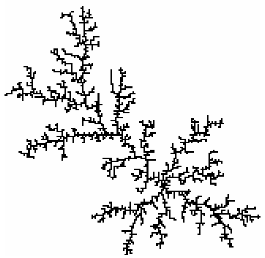
- The location of the entry point of the new particles is an important consideration, for the time taken to render a single tick is proportional to the distance from that location to wherever it finally lands.
- The use of indexing of the approximate location of points can significantly speed up the algorithm, as it prevents the tick time increasing exponentially.

Algorithmic Considerations

- When dealing with the set of transforms, if you add (a, b), it is important to also add (-a, -b) so that we allow sufficient randomness, while allowing the attraction force towards the centre to be the primary influence on the particle.
- This is also done because, through investigation, it appears that, without this provision, there are a significant number of cases where the particle never gets close enough to the existing cluster.

Symmetry





- Snowflakes in nature are six sided, and have plane and rotational symmetry.
- In order to enforce this in a DLA algorithm, it is necessary to create matching points for every point that is added to the cluster.

Empirical Analysis of Efficiency

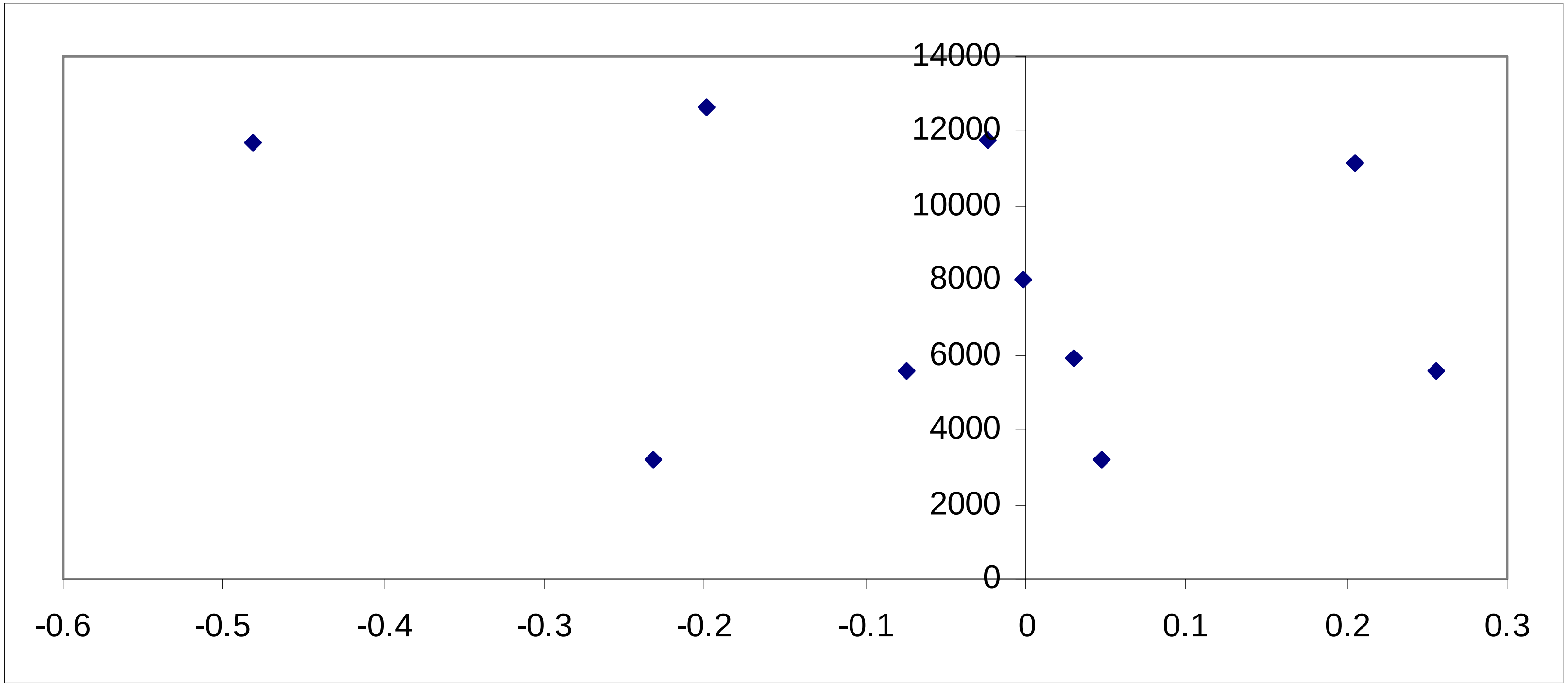
- The efficiency of the algorithm to generate DLA snowflakes depends on the tendency of the particles to gravitate towards the centre of mass.
- *Hypothesis*: if x_{av} and y_{av} represent the average x and y components of the transforms being used, then sets of transforms where $x_{av} + y_{av}$ is close to zero should be the most efficient, as the overall effect of the set of transforms will be to move the particle directly along $y = -x$ towards the origin (they start at coordinates (-150,150)).

Empirical Analysis of Efficiency

The table lists the number of steps needed to move from the corner to the cluster. All values are taken as an average over 200 particle launches.

Analysis 1					
Movement avg	Sample 1	Sample 2	Sample 3	Avg	SEM
-0.231909427	3185.915	3240.69	3219.39	**3215.332**	**0.86%**
-0.023717358	11721.09	11371.63	12258.38	**11783.7**	**3.79%**
0.029775505	5991.83	5874.005	5896.305	**5920.713**	**1.06%**
-0.199321596	12838.11	12897.47	12220.08	**12651.89**	**2.97%**
0.04719946	3197.325	3170.67	3192.14	**3186.712**	**0.44%**
-0.002134458	8135.505	7793.87	8117.27	**8015.548**	**2.40%**
-0.481326842	12053.13	11477.19	11608.46	**11712.93**	**2.58%**

Empirical Analysis of Efficiency



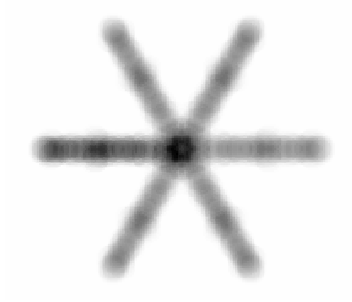
Evidently random, but consistent!

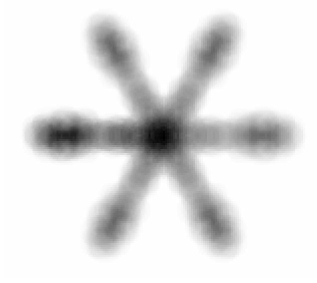
Empirical Analysis of Area

- Because these particular DLA snowflakes are made up of individual point particles, it is impossible to work out the area of the snowflake. Even if we assume that the particles have a particular radius, the area to be calculated would be significantly complicated by overlapping components
- As such, we melt the snowflake.

Melting the snowflake

- Melting the snowflake not only makes counting area easier and more consistent with intuitive notions, but they also look more like real snowflakes, which do not have the same level of intricate detail as those generated by a pure DLA algorithm.

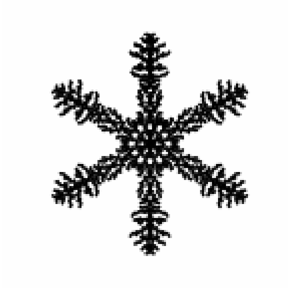
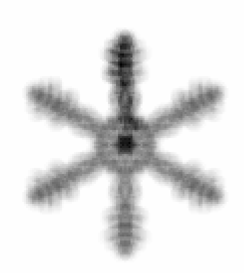
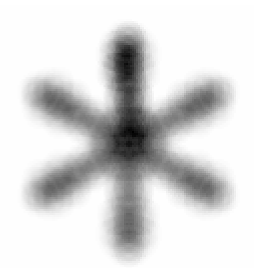




Melting the snowflake

- “Melting” is done on computer by taking a moving average of the number of points in a certain area.
- For each pixel in the diagram, it is coloured according to how many points lie within a certain radius of that pixel.
- The size of the snowflakes, expressed in pixels, would not be equal for all snowflakes. It is dependent upon its shape and distribution.

Melting the snowflake

Original	Smudge factor = 4	Smudge factor = 8
		

Empirical Analysis of Melting

- Questions: Is the area of the melted shape related to the bias in the set of transforms (just like we did for the previous example)? Is the area of the melted shape related to the general shape of the figure?

Empirical Analysis of Melting

(Movement avg)	Absolute value	Sample 1	Sample 2	Sample 3	Avg	SEM	
-0.24144867	0.24144867	5709	5835	5627	**5723.667**	**1.83%**	**Tendrils**
-0.317337106	0.317337106	4697	4754	4841	**4764**	**1.52%**	**Round**
-0.14723883	0.14723883	5381	5705	5669	**5585**	**3.18%**	**Tendrils**
0.180771687	0.180771687	5865	5119	5299	**5427.667**	**7.17%**	**Tendrils**
-0.09802246	0.09802246	4907	5533	4593	**5011**	**9.55%**	**Round/Tendril**

- Clearly from the table, there is no relationship between the overall positive or negative bias in the transforms and the shape of the snowflake.
- However, like with the previous example, the fact that the values generally coincide with each other suggests that the data exhibits features of a chaotic system. A small change in one variable can lead to substantial differences in the final outcome.

Dimension of a DLA snowflake

- The dimension of a DLA snowflake *cannot* be revealed using the method discussed last week of comparing the scaling between different levels of magnification because, by definition, the randomness of DLA ensures that it does not exhibit strict self-similarity.
- Hence the "box counting dimension" is replaced by the "mass dimension".

Dimension of a DLA snowflake

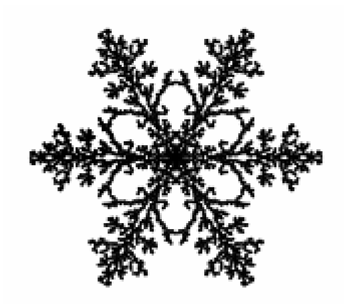
- If N(r) denotes the number of particles in a circle (or sphere) of radius r, then for large r we expect

$$N(r) = k.r^d$$

  for some constant k and for $d = d_m$.
- http://classes.yale.edu/fractals/Panorama/Physics/DLA/DLA6.html suggests that we should get a value of around $d_m = 1.71$ for clusters in the plane.
- Let's see what we get!

Dimension of a DLA snowflake

- Using data from the program to examine the number of particles that lie within a certain distance of the origin:



Dimension = 1.48

Dimension of a DLA snowflake

Radius	Number of Particles	Log(radius)	Log(Particles)
5	105	0.69897	2.021189299
10	300	1	2.477121255
15	547	1.17609126	2.737987326
20	833	1.30103	2.920645001
25	1132	1.39794001	3.053846427
30	1431	1.47712125	3.155639634
35	1730	1.54406804	3.238046103
40	2263	1.60205999	3.354684554
45	2835	1.65321251	3.452553063
50	3342	1.69897	3.524006446

Dimension of DLA snowflake

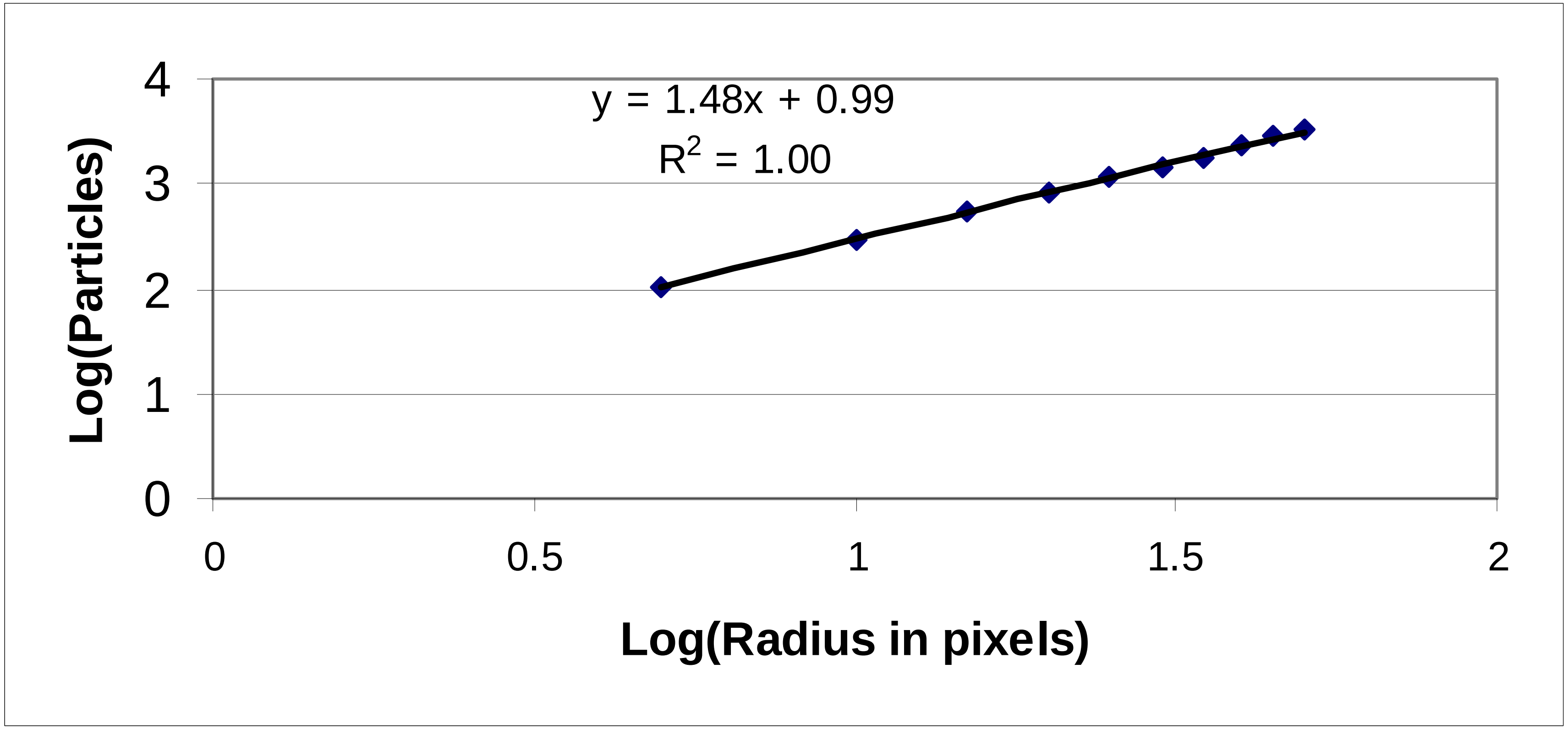
$$n = k.r^d$$

$$\log n = \log(k.r^d) = d \log r + \log k$$

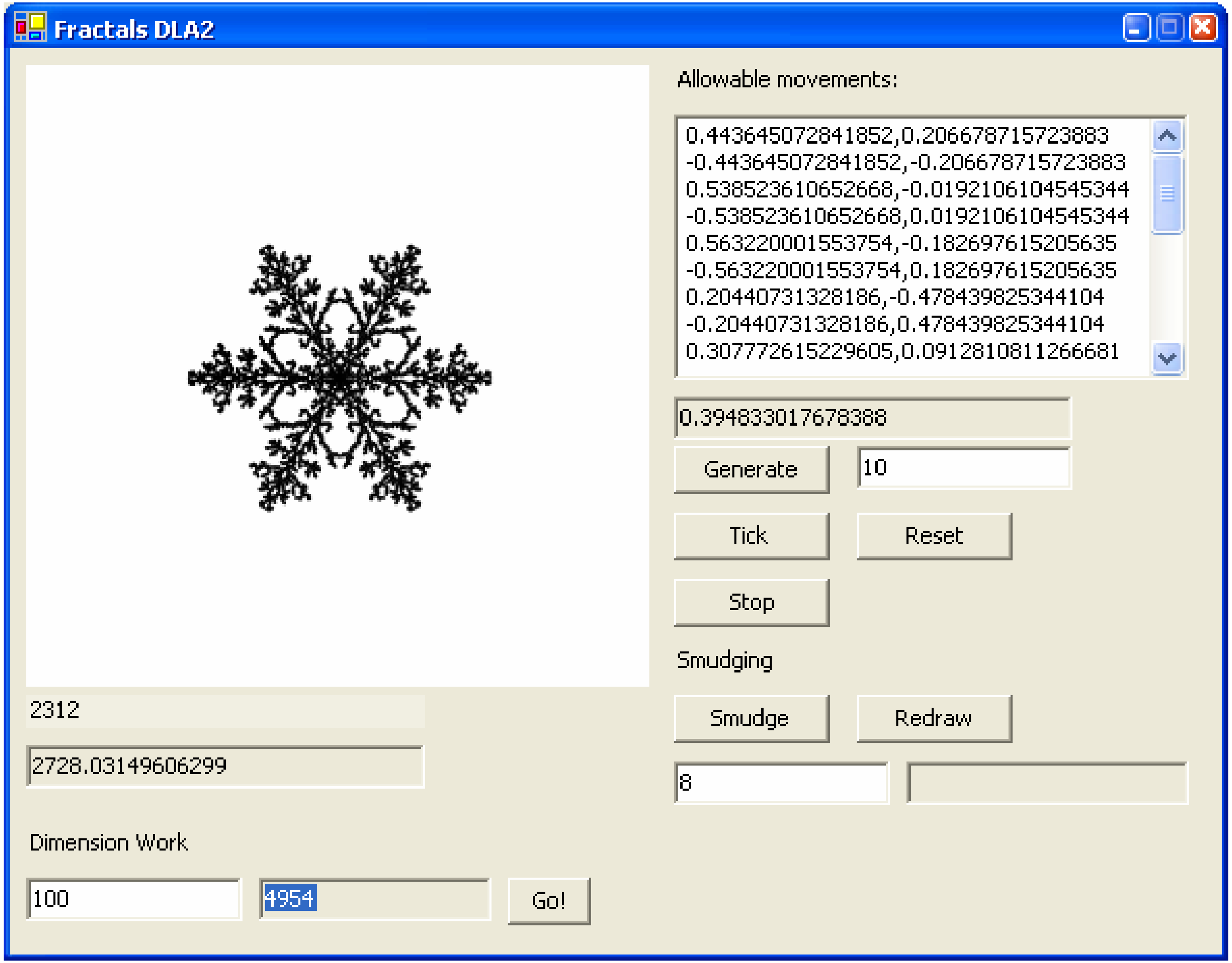
$$c.f.: y = mx + b$$

Hence, the gradient of the line in the log-log graph will give us the dimension of the DLA snowflake.

Dimension of a DLA snowflake



Therefore, gradient is 1.48.


Fractals DLA2
Allowable movements:
0.443645072841852,0.206678715723883
-0.443645072841852,-0.206678715723883
0.538523610652668,-0.0192106104545344
-0.538523610652668,0.0192106104545344
0.563220001553754,-0.182697615205635
-0.563220001553754,0.182697615205635
0.20440731328186,-0.478439825344104
-0.20440731328186,0.478439825344104
0.307772615229605,0.0912810811266681
0.394833017678388
Generate
10
Tick
Reset
Stop
Smudging
Smudge
Redraw
8
2312
2728.03149606299
Dimension Work
100
4954
Go!