

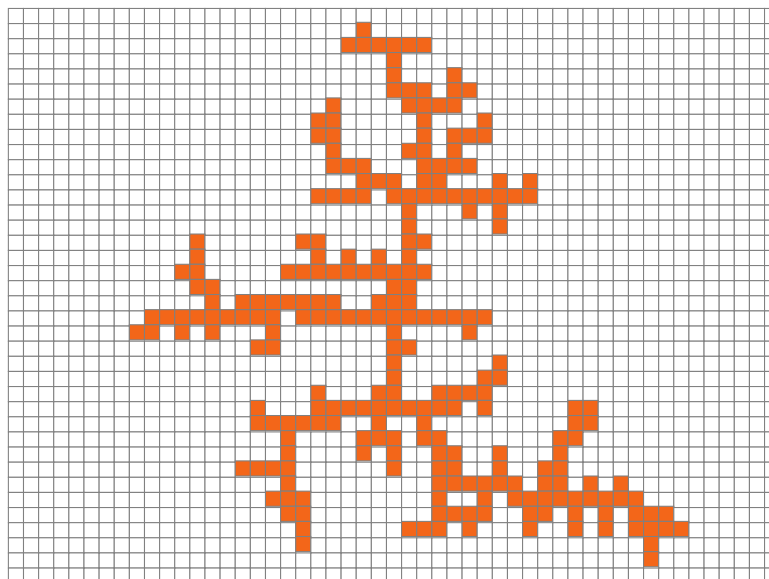
1. Week 4: DLA

1.1 Diffusion Limited Aggregation



https://en.wikipedia.org/wiki/Diffusion-limited_aggregation

In its simplest form, DLA occurs on a grid of square cells. The cell at the center of the circle is the location of the seed point, a particle stuck at that square. Now pick a square on the perimeter of the grid and place a wandering particle on that square. At each iteration, this particle moves to one of the four adjacent squares, left, right, above, or below. When a wandering particle arrives at one of the four squares adjacent to the seed, it sticks there forming a cluster of two particles, and another moving particle is released. When a moving particle arrives at one of the squares adjacent to the cluster, it sticks there.



Note that the grid is toroidal - this means that if a particle goes off the top of the grid, it reappears at the bottom. Likewise, if it goes off the edge, it will appear on the other side.

Exercise 1.1 This file codes empty cells as a space and occupied cells as a hash symbol (#). Display the board after 250 particles have been added, using plain text. You may assume that the grid is always 50 cells by 50

Make sure all your code is fully ANSI compliant, and fully follows the house-style guidelines. ■

90%

1.2 ncurses

C has no inherent functionality to allow printing in colour etc. Therefore, a programming library known as `ncurses` was created in 1993 to allow terminals to interpret certain control-codes as colours and other effects.

The library itself is somewhat complex, allowing keyboard and mouse events to be captured and a whole range of simple graphics functionality. On the web page is my ‘wrapper’ for the library, along with a program demonstrating its use. This will only work in unix-style terminals. Note that after you begin `ncurses` mode (using `Neill_NCURS_Init()`) that you can’t print to `stdout` or `stderr`, until you switch it off (using `Neill_NCURS_Done()`).

To compile the code you’ll have to use both my code `neillncurses.c` and also link in the `ncurses` library. A typical compile might look like

```
gcc yourcode.c neillncurses.c -Wall -Wfloat-equal -Wextra -O2  
-pedantic -ansi -lncurses -lm
```

Exercise 1.2 Adapt the `dla` code so that the output is displayed using this library, with occupied cells being red and empty being black. The main loop will update the board, display it, and repeat until a quit event occurs (e.g. a mouse click or the `ESC` key is pressed). ■

10%