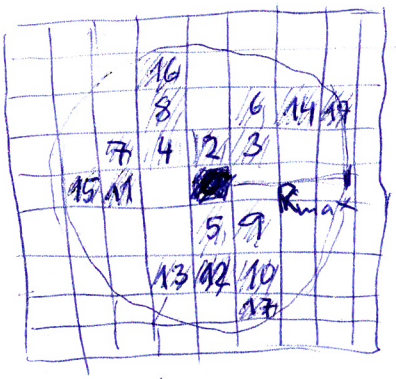
Diffusion-limited aggregation (DLA)

- a problem where MC simulation can be used to generate complex structures from simple rules
  - the evolution of the system is modelled as a sequence of random states

- example:



we start in the middle and then we repeat a random movement in the 2d lattice until we hit the site next to an occupied site → deposition

  - the resulting structure is sometimes called Brownian tree
  - a model of the system which grows (eg. crystal growths)
    → direct modelling is complicated due to changing boundary conditions and instabilities
    => MC simulations
  - to compare a model and reality we have to determine some quantities which can be both measured and calculated
    e.g. fractal dimension - it can be estimated as

$$D_f \doteq \frac{\log N(a)}{\log L(a)}$$

where $N(a)$ is the number of particles in a box of side length a (occupied sites)
and $L(a)$ is the number of sites which form the edge of the box

  - for 2d DLA it should be about 1,7
    for 3d DLA about 2,5

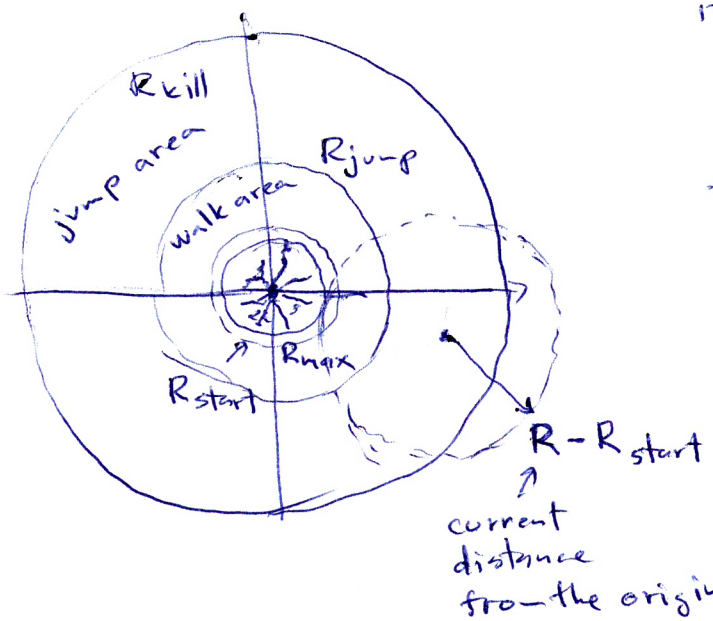
- basic algorithm

1) initialization - seed particle(s)
   - fill one or more sites to which randomly walking particles will attach
   - it can be a particle in the middle or a line along one of edges etc.

2) repeat the following until the size of the structure is large enough:

   a) launch a new particle
      - place a single particle with equal probability to a starting positions (e.g. on a circle with radius $R_{start} > R_{max}$ where R_{max} is a current maximum distance from the seed or on a line etc.)
      (note: R_{start} can change during simulation to save some time, it can be even $R_{start} = R_{max} + 1$ for the most efficient simulations)

   b) diffusion
      - move the particle randomly until its neighbour site is occupied or until it leaves a prescribed area ($R > R_{kill}$)
      - to do that efficiently we do a proper random walk only close to R_{max}, i.e. for $R < R_{jump}$
      => if $R > R_{jump}$ then we jump to a random site along a circle around the current position with a radius $R - R_{start}$
      (R_{jump} is adjusted during simulation to be between $R_{start} < R_{jump} < R_{kill}$)



c) aggregation – if one of the nearest neighbours of the walking particle is occupied, add this particle to the cluster

- increase the „mass" M of the cluster by one, if $M \geq M_{desired}$ then stop the simulation
- otherwise adjust R_{max} if necessary and accordingly R_{start} and R_{jump}

- more on the fractal dimension of the generated „Brownian tree"
  - in general, the box-counting dimension of a structure is defined as

$$D_{box}(S) = \lim_{\varepsilon \to 0} \frac{\log N(\varepsilon)}{\log \frac{1}{\varepsilon}}$$

(S: a set in $\mathbb{R}^n$)

where $N(\varepsilon)$ is the number of boxes of side length ε required to cover the set S

  - in our case, we have a fix size of the boxes (ε) (one square of the lattice), but we can generate a larger and larger Brownian tree if we generated sufficiently large structure we can take a box of the size $a \times a$ where $a = n\varepsilon$, thus $\frac{1}{\varepsilon} \sim n$ and instead of limit $\varepsilon \to 0$ we take a limit $n \to \infty$
  - thus we get an alternative definition

$$D_{box}(S) = \lim_{n \to \infty} \frac{\log N(n)}{\log n}$$

where $N(n)$ is the number of occupied sites of the lattice in a box (square) of size $n \times n$.