Parallel DLA
(Diffusion-limited aggregation)

מאת: דרור אהרוני

What is DLA?

- **DLA** is the process whereby particles undergoing a random walk due to Brownian motion cluster together to form aggregates of such particles
- can be observed in many systems such as electro deposition, Hele-Shaw flow, mineral deposits, and dielectric breakdown.

Purposes

- **exploitation of oil resources** embedded in sand by means of water pressed into the sand. So water-oil-surface may or may not exhibit shapes which resemble those of DLA
- **catalyst** forming - These materials of great value in chemical engineering may also show strongly branched surfaces with many holes and channels. The very large surface area is often quite relevant for the catalytic process.

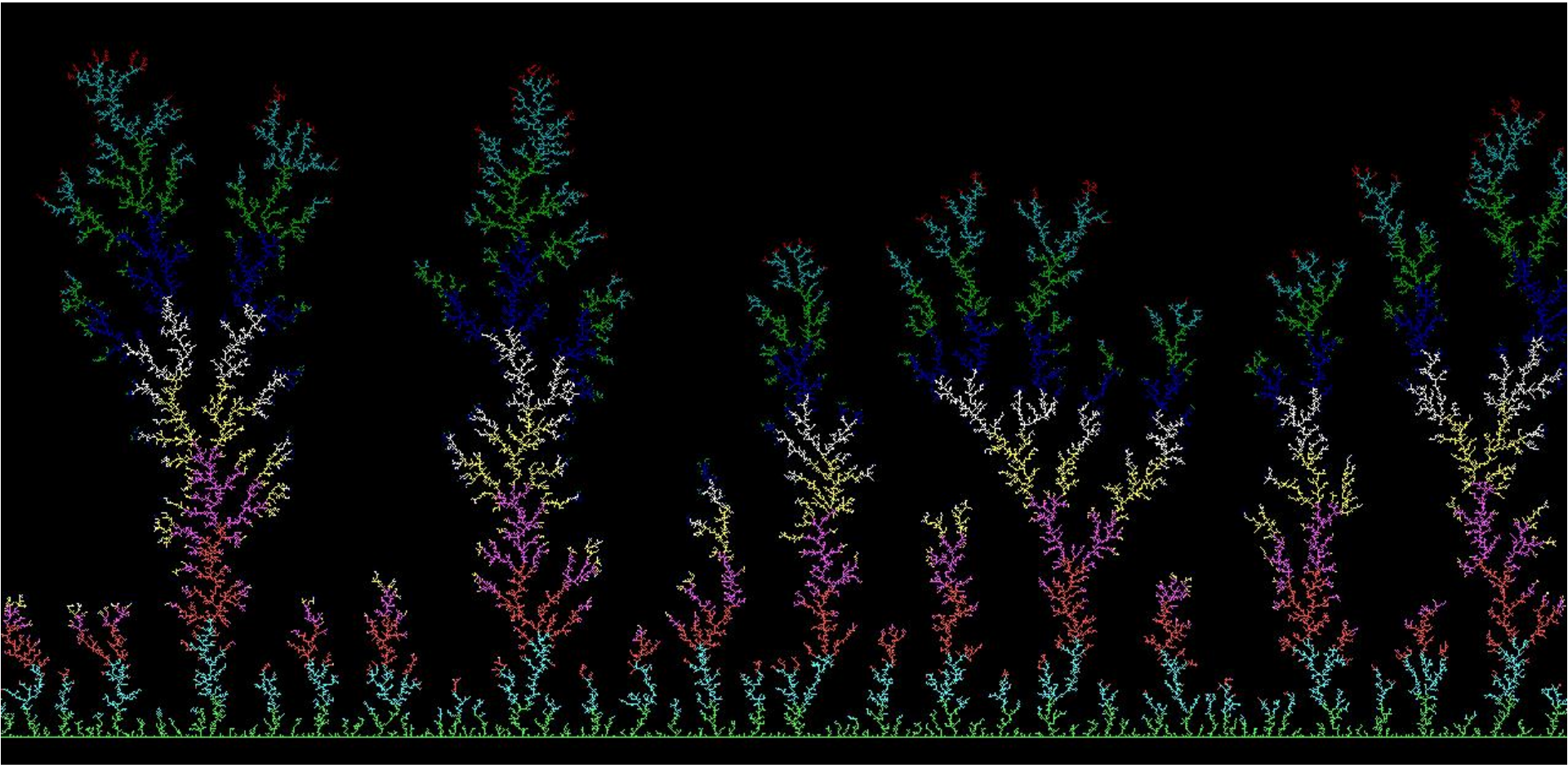
Purposes

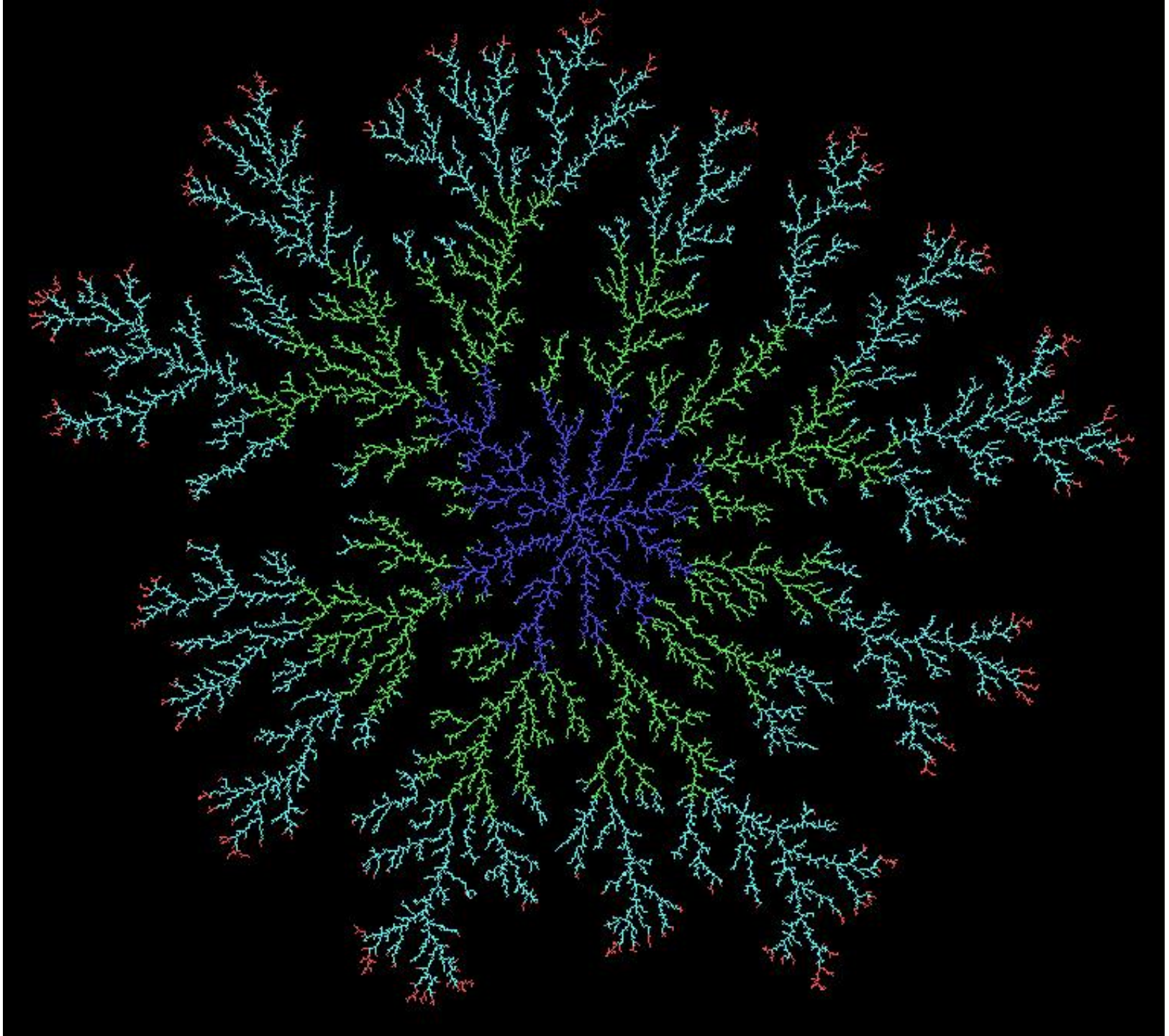
- When metals like copper or gold are separated electrolytically out of a solution (*electro deposition*), shapes like DLA-cluster may occur under certain conditions
- when metals are deposited onto surfaces from gaseous phase to get very thin films a DLA-growth might occur

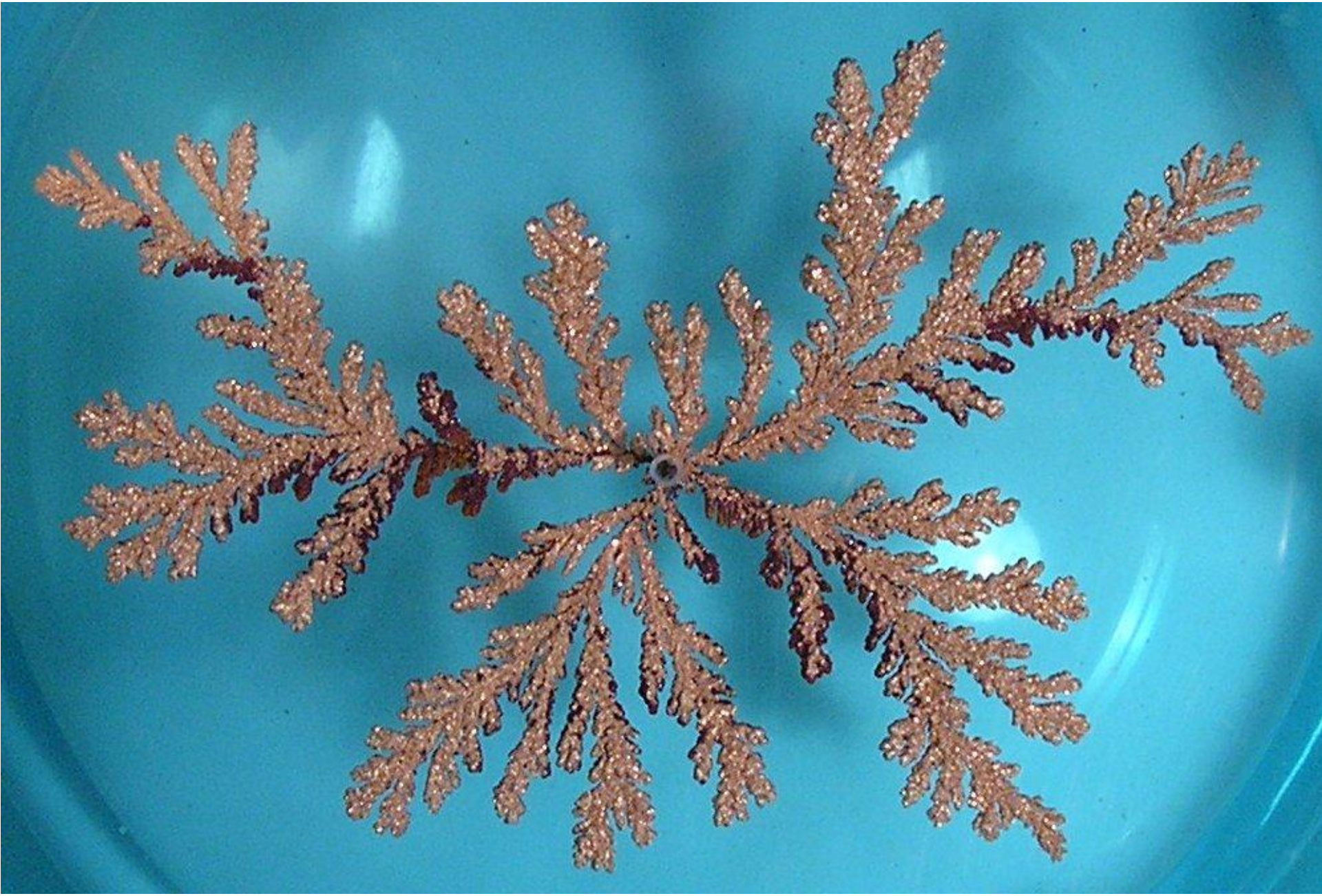
Purposes

- The generation of **polymers** out of solutions may follow a DLA-mechanism. The properties of the resulting molecules may depend strongly on this.
- The same holds when **solid phases** form in undercooled liquid phases.
- Another example for such a **deposit** is carbon on the walls of a cylinder of a *Diesel* engine. These effects may or may not be wanted, depending on the destination of the deposits.

דוגמאות







The process

- Seeding – one particle, a line of particles
- Where the seeding particle is placed – middle, top, bottom
- Deciding where is the starting point of the new particle – random (circle, anywhere), top boundary and so forth

The project

- Finding a way to make a parallel computing for a DLA process on a large grid while displaying real-time graphics

The difficulties

- Writing the initial program while trying to have a high “sticking” probability
- Finding a way to make the computation parallel and faster
- Adding real-time graphics of the process using parallel computation

Possible Solutions – writing

- For high ‘sticking’ probability I have set the starting point near the first seed (circle/line) and updated it when the growth got close to it

Possible Solutions – Paralleling

- Parallel computing over a SMM (shared memory multiprocessing) cluster, where all computer share the ‘lattice’ 2D array
- Parallel computing with MPI with ‘Master-Servant’ programming, where the ‘Master’ updates and distribute the ‘lattice’ 2D array [probably will be too slow]

Possible Solutions – graphics

- Currently writing the program with C.
- Might shift to another programming software which have a better graphics abilities