Talk on Mar 28 $\rightarrow$ Apr 4:
IDLA and Rotor-Router Motion with Drift

Ruojun

April 4, 2019

Today I would like to talk about this problem about IDLA and RR ...

Today I would like to introduce regarding the Internal Diffusion Limited Aggregation and Roter-Router motion. This is a project I worked with Professor Thiffeault. I learned a lot by working on this project and I would like to share it with you today.

1 Internal DLA

I started by considering a simulation for IDLA motion. IDLA represents for the Internal Diffusion Limited Aggregation. It describes the random walk of particles with a single source at origin.

1.1 Definition w/ the description of the simulation

(Dra.) The plane starts empty. In the code we create such a plane with N*N grid points. Particles are added to the origin one at a time, each particle occupies the first empty site it reaches. A particle at an occupied site walks to a random neighbor, each with probability 1/4 (Kleber, 2005, p.60). If the grid is unoccupied, the particle would then settle down there; if it is occupied, it would further walk randomly to find the next unoccupied grid.

For example, (draw) ...

1.2 Run live (intdla.m)

(Sim.) The simulation shows the IDLA motion of the 1000 particles in total added at the origin.

An observation here is: It converges to a circle as $N^{1/2}$. Such a correlation between ... would also be obs. later for other simulations.

2 Rotor-router

Then I studied another simulation where particles do not perform random walks. Last spring in this seminar Professor Thiffeault introduced a journal called by "Goldbug Variations" written by Michael Kleber (write MK, 2005) in which he described this motion called roter-router. In the article Kleber used "bugs" instead of "particles".

2.1 Definition

(Dra.) Basically at first we also have an empty plane. Bugs still get added repeatedly at the origin. Each lattice site is equipped with an arrow, or rotor, which can be rotated so that it points at any one of the four neighbors. The arrows are all pointed to the North at first. These arrows would later determine how the bugs move on the plane.

The first bug is added at the origin and occupies it forever. And it sets the arrow there pointing to the East. The second bug arriving at the origin rotates the rotates the arrow one quarter counterclockwise. Now the arrow points to South. Then it moves one step to South. If this lattice is unoccupied, it would then settle down there.

The general rule is: Any bug arriving at an occupied site rotates the arrow one quarter counterclockwise, and then moves to the neighbor at which the rotor now points–where it may find an empty site to inhabit, or it may find a new arrow directing its next step (Kleber, 2005, p.57).

2.2 Show some simulations (or run live) (rr2d.m)

(Sim.) The simulation shows the RR motion of the 1000 bugs in total added at the origin, where black represents the empty site; the color very close to black gives the site where an arrow pointing to East locates; light gray is for the arrow pointing to East; white is for the arrow pointing to South; dark gray is for the arrow pointing to West.

(7 minutes)

3 IDLA & rotor-router with drift

Based on these two simulations, I studied the motion of the particles after I add a drift to each simulation and tried comparing the shape of the occupied region.

3.1 IDLA with drift

(Dra.) The particle are still added to the origin, and a particle would move randomly towards a neighbor of 4 move up with one step to either settle down in an unoccupied place or to move randomly again until it finds an unoccupied grid point.

3.2 intdla_drift.m (no scaled)

(Sim.) I added a vertical drift in both cases.

Q: blank grids? A: Only move vertically with even steps

4 Show "modified rotor-router" still agrees with random simulations ?

4.1 RR with drift

The bug would move and rotate the arrow in one grid point as usual and move to the north with one step to settle down in an unoccupied place or it would continue to move until it finds an unoccupied place.

4.2 rr2d_drift.m (no scaled)

5 The vertical grows as $N^{2/3}$

We did the rescaling for ...: We tried to studied how it shape grows with drift and the correlation between the shape and the number of ...

In order to study the pattern of the growth, I started by finding the factor that causes the vertical growth. (mma.) I recorded the total number of the particles/bugs and the maximum y coordinate the shape reaches. I fit the data with the function $f = aN^b + c$; we find that $b \approx .66$ numerically.

The height grows as $N^{1/3}$.

5.1 Shape of the Boundary of IDLA (intdla_driftC.m)

We used the empirical formula obtained from above and tried in the simulation to construct a plane where the IDLA and roter-router patterns could well locate in the center:

```
maxy = ceil(1.28164*Npart^.66 + 4.96289); maxx = 3*ceil(Npart/maxy);
```

5.2 Shape of the Boundary of RR case (rr2d_driftC.m)

```
gridquadsize = @(Nbugs) ceil(.6*sqrt(Nbugs));
```

We find that in both cases the shapes the vertical maximum grows as $N^{2/3}$.

(5 min) (12)

6 Point source with drift

To study the two cases better, one way is to examine a limiting behavior. That is, we consider a grid with infinite number of grid points. We could consider an equation describing point source with drift.

6.1 Find steady Green's function

Now we consider an equation that describes a concentration $\psi(\mathbf{r})$ with diffusion and a source added at the origin

$$U\,\partial_y\phi = D\,\Delta\phi + \delta(\boldsymbol{r}) \tag{6.1}$$

Let $\psi(\boldsymbol{r}) = \mathrm{e}^{-Uy/2D}\phi(\boldsymbol{r})$; then ψ satisfies

$$-\,D\,\Delta\psi + \frac{U^2}{4D}\,\psi = \delta(\boldsymbol{r}). \tag{6.2}$$

This is a Hemholtz equation. A benefit of this equation is that now we have a symmetric solution with respect to r. For convenience we define the inverse length scale $a = U/2D$:

$$-\,\Delta\psi + a^2\,\psi = \delta(\boldsymbol{r})/D. \tag{6.3}$$

The solution to this must be dependent on r only, so we try to solve:

$$\frac{1}{r}\frac{d}{dr}\left(r\frac{\partial\psi}{\partial r}\right) - a^2\psi = 0. \tag{6.4}$$

This is a modified Bessel equation of the second kind; the solution that decays as $r \to \infty$ is

$$\psi(r) = c_1\,K_0(ar). \tag{6.5}$$

The Green's function we seek is then of the form

$$\phi(\boldsymbol{r}) = \mathrm{e}^{ay}\,\psi(r) = c_1\,\mathrm{e}^{ay}\,K_0(ar). \tag{6.6}$$

_____ (below)

To find the constant a, we observe that for small x

$$K_0(x) \sim -\log x, \qquad x \to 0. \tag{6.7}$$

Hence, for the Green's function to limit to that for the Laplacian with a point source of strength $-1/D$ as $r \to 0$, we require

$$\phi(\boldsymbol{r}) \sim -c_1 \log r \sim -\frac{1}{2\pi D} \log r, \qquad r \to 0. \tag{6.8}$$

_____ (above)

We obtain finally the Green's function

$$\phi(\boldsymbol{r}) = \frac{1}{2\pi D} \, \mathrm{e}^{ay} \, K_0(ar), \qquad a = \frac{U}{2D}. \tag{6.9}$$

A useful asymptotic form for this is

$$\phi(\boldsymbol{r}) \sim \frac{\mathrm{e}^{a(y-r)}}{\sqrt{4\pi DUr}}, \qquad r \to \infty. \tag{6.10}$$

In particular, this has very different limits along the $x = 0$ axis dependent on whether we are upstream ($y \to -\infty$) or downstream ($y \to \infty$):

$$\phi \sim \frac{1}{\sqrt{4\pi DU|y|}} \begin{cases} 1\,, & y \to \infty; \\ \mathrm{e}^{-2a|y|}\,, & y \to -\infty. \end{cases} \tag{6.11}$$

(10 min) (22)

6.2 Shape of contours

We then tried to estimate the contour with the Green function we obtained and see if it matches the simulation.

(???) To find the shape of large contours $\phi(\boldsymbol{r}) = c/\sqrt{4\pi DU} > 0$ we use the asymptotic form Eq. (6.10):

$$\mathrm{e}^{a(y-r)} = c\sqrt{r}. \tag{6.12}$$

The furthest extent of the contour for $y > 0$ is obtained by setting $r = y = y_{\max}$ and then solving

$$1 = c\sqrt{y_{\max}} \quad \Longleftrightarrow \quad y_{\max} = 1/c^2. \tag{6.13}$$

_____ (below)

We let furthest extent of the countour for $y < 0$ be $y = -y_{\min}$. Assuming that $y_{\min}$ is large and that Eq. (6.10) is still valid, we set $r = -y = y_{\min}$ and then solve

$$\mathrm{e}^{-4ay_{\min}} = c^2\, y_{\min}. \tag{6.14}$$

The solution to this is given in terms of the Lambert W-function as

$$y_{\min} = \frac{1}{4a}\, W(4a/c^2). \tag{6.15}$$

For small c, this has asymptotic expansion

$$y_{\min} \sim \frac{1}{4a}\left(\log(4a/c^2) - \log\log(4a/c^2)\right), \qquad c \to 0. \tag{6.16}$$

This is indeed getting slowly larger as $c \to 0$, consistent with using the approximation Eq. (6.10), but the extent of the contour in y is completely dominated by $y_{\max}$. We can thus safely use $y_{\min} = 0$ when computing the area.

_____ (above)

$y_{min} \approx 0$ due to exponential decay on this side.

Now that we know the limits in y, to compute the area as a function of y we use Eq. (6.12)

$$\frac{\mathrm{e}^{a(y-\sqrt{x^2+y^2})}}{(x^2+y^2)^{1/4}} = c \tag{6.17}$$

and try to solve for x. (We are not sure which contour to take exactly but we could figure this out by mass conservation.)

We rescale $X = \sqrt{a}\, c\, x$, $Y = c^2 y$, and then Taylor expand:

$$\frac{\mathrm{e}^{-X^2/2Y}}{\sqrt{Y}} + \mathrm{O}(c^2) = 1. \tag{6.18}$$

Figure 6.1: Numerical contours (solid) compared to the approximation (eq:xapprox).

Solving for X, we find for the shape of the contour

$$X = \sqrt{Y \log Y^{-1}} + \mathrm{O}(c^2), \qquad 0 < Y < 1, \tag{6.19}$$

or in terms of the unscaled variables,

$$x_{\max}(y) = \sqrt{(y/a) \log (c^2 y)^{-1}} + \mathrm{O}(c), \qquad 0 < y < c^{-2}. \tag{6.20}$$

Now we can compute the estimated area as

$$A = \frac{2c^{-3}}{\sqrt{a}} \int_0^1 X(Y)\, \mathrm{d}Y = \frac{2}{3} \sqrt{\frac{2\pi}{3a}}\, c^{-3}. \tag{6.21}$$

The crucial observation is that $A \sim c^{-3}$. Hence,

$$y_{\max} \sim c^{-2} \sim A^{2/3} \tag{6.22}$$

as we find in the simulations. Figure fig:contours compares these approximate contours to numerical simulations.

(*** Note that for us the area is twice the number of particles, since we have a checkerboard pattern. Also, I think $a = 1$ (We used $a \approx .39$ in the code; we found this empirically.) is the right scaling, since $UT = 1$ gridpoint, and $\sqrt{2DT} = 1$ gridpoint as well.)

(We have not examined a and UT analytically; we just assign value to them empirically, which matches the simulation.)

6.3 intdla_driftB.m, rr2d_drift.m

(15 min) (37)

7 Small discrepancy: is it real or not?

Maybe the approximation could be improve by doing Stefan Problem.

8 Stefan problem

To describe the IDLA motion more precisely, we need to introduce Stefen problem, which has a time-dependent boundary condition.
We have an equation which could describe the IDLA motion with a source at the origin

$$\partial_t\, p = D\nabla^2 p + \delta(\mathbf{x})S \tag{8.1}$$

where S is a constant. In a region $\Omega(t)$ with the boundary $\partial\,\Omega = \{\mathbf{d}(t)\}$.

Consider a Dirichlet boundary condition

$$p|_{\partial\,\Omega} = 0. \tag{8.2}$$

Also,

$$\hat{\mathbf{n}} \cdot \dot{\mathbf{d}} = -\alpha\, \hat{\mathbf{n}} \cdot \nabla p \tag{8.3}$$

where D and α are constants.

(5 min) (42)

_____ (below)

8.1 Solve for the circular case

8.1.1 Find $R(t)$

For the circular case, solve for $p(r,t)$, with a source at origin $\delta(\mathbf{x})$

$$\partial_t p = \frac{D}{r}\frac{\partial}{\partial r}(r\frac{\partial p}{\partial r}) + \delta(\mathbf{x})S \tag{8.4}$$

From (7.3) and (7.5), the boundary conditions are

$$p(R(t),t) = 0 \tag{8.5}$$

$$\dot{R} = -\alpha\frac{\partial p}{\partial r}(R(t),t) \tag{8.6}$$

Consider the integral for mass

$$M(t) = \int_{\Omega(t)} p \, dV = 2\pi \int_0^{R(t)} p(r,t) r dr \tag{8.7}$$

By the conservation of mass

$$\dot{M}(t) = 0 \tag{8.8}$$

$$\dot{M} = 2\pi [\int_0^{R(t)} p_t r dr + p(R(t),t) R\dot{R}] \tag{8.9}$$

$$= 2\pi \int_0^R [D\partial_r (r\partial_r p) + Sr\delta(\mathbf{x})] dr \tag{8.10}$$

$$= 2\pi D [r\partial_r p]_0^{R(t)} + S \int_{\Omega(t)} \delta(\mathbf{x}) dV \tag{8.11}$$

$$= 2\pi DR\partial_r p(R(t),t) + S \tag{8.12}$$

With (7.8),

$$\dot{M} = 2\pi DR(-\frac{\dot{R}}{\alpha}) + S \tag{8.13}$$

$$= -\frac{D}{\alpha}\frac{d}{dt}(\pi R^2) + S \tag{8.14}$$

$$= -\frac{D}{\alpha}\dot{A} + S \tag{8.15}$$

where

$$A(t) = \pi R^2(t) \tag{8.16}$$

and with (7.10)

$$\dot{A} = \frac{S\alpha}{D} \tag{8.17}$$

Hence,

$$M(t) = -\frac{D}{\alpha} A(t) + St \tag{8.18}$$

$$A(t) = (\frac{S\alpha}{D})t = \pi R^2(t) \rightarrow R(t) = \sqrt{\frac{S\alpha}{\pi D} t} \tag{8.19}$$

8.1.2 Method 1: Find Green's function (might not work)

?

Consider (7.5) and a green's function $G(r)$ that satisfies

$$D\nabla^2 G(r) = \frac{D}{r}\partial_r(r\partial_r G(r)) = -S\delta(\mathbf{x}) \tag{8.20}$$

Hence,

$$G(r) = c\log(r) \tag{8.21}$$

where

$$\nabla^2 \tilde{G} = -\delta(\mathbf{x}) \tag{8.22}$$

$$\tilde{G} = -\frac{1}{2\pi}\log(r) \tag{8.23}$$

where $\tilde{G}$ is the Fourier Transform of $G(r)$ and $c = -\frac{\delta(\mathbf{x})}{2\pi D}$ (?)
Let $\tilde{p} = p - G(r)$,

$$\partial_t \tilde{p} = \frac{D}{r}\frac{\partial}{\partial r}(r\partial_r \tilde{p}) \tag{8.24}$$

such that $\tilde{p}(R(t), t) = -G(R(t))$.

Solve the equation without the source term (might not work) Consider $\tilde{p} = \theta(t)\phi(r)$. With separation of variables,

$$\frac{d_t\theta\phi}{\theta\ \phi} = \frac{D}{r}\frac{\theta\partial_r(r\partial_r\phi)}{\theta\ \phi} = -\lambda \tag{8.25}$$

Hence,

$$\theta(t) = c_1 e^{-\lambda t} \tag{8.26}$$

And

$$\phi(r) = c_2 J_0(r\sqrt{\lambda_{0n}}) \tag{8.27}$$

Consider the boundary condition $\tilde{p}(R(t), t) = -G(R(t)) = c\log(R((t))$. We could also write $\tilde{p}(R(1), 1) = -G(R(1)) = c\log(R((1)) = c\log(\sqrt{\frac{S\alpha}{\pi D}})$. Thus,

$$\tilde{p}(r, t) = c_3 e^{-\lambda_{0n} t} J_0(r\sqrt{\lambda_{0n}}) \tag{8.28}$$

where

$$\tilde{p}(R(1),1) = c_3 e^{-\lambda_{0n}} J_0(\sqrt{\frac{S\alpha}{\pi D}}\sqrt{\lambda_{0n}}) \tag{8.29}$$

$$= c\log(\sqrt{\frac{S\alpha}{\pi D}}) \tag{8.30}$$

Solve for λ_{0n}, ... ?

$$p(r,t) = c_3 e^{-\lambda_{0n}t} J_0(r\sqrt{\lambda_{0n}}) + c\log(r) \tag{8.31}$$

Consider the initial condition $p(r,0) = 0$,

$$p(r,0) = c_3 J_0(r\sqrt{\lambda_{0n}}) + c\log(r) \tag{8.32}$$

...

Do I need to provide a handout or notes in the end?

8.1.3 Method 2: Rescale p