In-Class Work: Random Walk

1. Random Walk in Two Dimensions

Our goal is for today to write a program for a random walk. We focus here on the random walk in two dimensions (instead of one dimension), because the random walk in two dimensions will be part of the fractal growth model, which we will program in the following classes.

1a. Flow Chart

Work in groups of two and draw a flow chart for a program of a random walk in two dimensions for N timesteps, which prints out at each time step t, x, y. Be as detailed as possible.

1b. Program

Write a program for a random walk in two dimensions for $N = 50000$ timesteps. As a start copy into your working directory the random number generator sample program
~kvollmay/sunhome/classes.dir/capstone_s2008.dir/traffic.dir/float_rand0-1.cc
Change the program such that it print on screen for each time step t,x and y, so you want to end up with three columns and 50000 lines. Have a look at $x(t)$ and $y(t)$ with
`executable | xgraph -m` and
`executable | gawk '{print $1,$3}' | xgraph -m`

1c. Movie

Next let's make a movie of your random walk. Define a lattice of size 100x100 and initialize it with all sites equal to zero. Put your initial walker at $x = 50$ and $y = 50$. For each timestep print the lattice such that you can pipe your output into DynamicLattice, that means that for each time step you print on screen the whole lattice (100x100) and one empty line in between whole lattices. Set the current lattice site (x, y) equal to 2 and all sites which were visited in the past equal to 1. You should obtain a white polymer with a red tail and blue background. Run your program for about 300 timesteps. To look at the movie you use
`executable.out | DynamicLattice -nx 100 -ny 100 -matrix`

2. Random Walk in One Dimension (Advanced)

Next we will do important analysis on the random walk. To simplify the task (and yet being able to get the main concept) let us use the random walk in one dimension. Before you start with 2a, please get me. I would like to discuss the following analysis with you.

2a. Program the random walk in one dimension, initialize with $x(0) = 0$ and plot $x(t)$. Use `NSTEPS = 5000` time steps.

2b. The next step is a preparation for 2c. Instead of printing every time step, print only once after N =`NSTEPS` time steps the resulting $x(N)$ and $(x(N))^2$.

2c. Now add a loop over simulation runs to your program from 2b. Each simulation run gives you an x and an x^2. Take the average over `NSIMRUNS=10000` simulation runs of x and an x^2 and print out the resulting averages $\langle x \rangle$ and $\langle x^2 \rangle$.

2d. Next no longer use the number of steps `NSTEPS` as constant but instead add a loop over N=`NSTEPS` to your program of 2c. For each `NSTEP` print out N=`NSTEPS` and $\langle x^2 \rangle$. Look at the resulting $\langle x^2(N) \rangle$.

In-Class Work: Fractal Growth (Newcomers)

1. Use your program from last class of 1b. or if it was not finished, then copy into your working directory
~kvollmay/sunhome/classes.dir/capstone_s2008.dir/fractal.dir/rand1b.cc

1c. Movie

Next let's make a movie of your random walk. Define a lattice of size 100x100 and initialize it with all sites equal to zero. Put your initial walker at $x = 50$ and $y = 50$. For each timestep print the lattice such that you can pipe your output into DynamicLattice, that means that for each time step you print on screen the whole lattice (100x100) and one empty line in between whole lattices. Set the current lattice site (x, y) equal to 2 and all sites which were visited in the past equal to 1. You should obtain a white polymer with a red tail and blue background. Run your program for about 300 timesteps. To look at the movie you use
`executable.out | DynamicLattice -nx 100 -ny 100 -matrix`

3. DLA: Circle and Random Walk Copy into your working directory
~kvollmay/sunhome/classes.dir/capstone_s2008.dir/fractal.dir/DLA3_sample.cc
The program includes already the initialization of the lattice and the starting point on a circle. Include in the program (at indicated place) the random walk being done by the particle which starts on the circle. Update the lattice for each random walk step, so that the random walker is visible as a red particle (`lattice[x][y]=2;`) on the blue background with the white seed in the middle. Look at your resulting lattice movie with
`executable | DynamicLattice -nx 100 -ny 100 -matrix`

In-Class Work: Fractal Growth (Advanced)

2. Use your program from last class of 2a. or if it was not finished, then copy into your working directory
~kvollmay/sunhome/classes.dir/capstone_s2008.dir/fractal.dir/rand2a.cc

2b. The next step is a preparation for 2c. Instead of printing every time step, print only once after N =`NSTEPS` time steps the resulting $x(N)$ and $(x(N))^2$.

2c. Now add a loop over simulation runs to your program from 2b. Each simulation run gives you an x and an x^2. Take the average over `NSIMRUNS=10000` simulation runs of x and an x^2 and print out the resulting averages $\langle x \rangle$ and $\langle x^2 \rangle$.

2d. Next no longer use the number of steps `NSTEPS` as constant but instead add a loop over N=`NSTEPS` to your program of 2c. For each `NSTEP` print out N=`NSTEPS` and $\langle x^2 \rangle$. Look at the resulting $\langle x^2(N) \rangle$.

3. DLA: Circle and Random Walk Copy into your working directory
~kvollmay/sunhome/classes.dir/capstone_s2008.dir/fractal.dir/DLA3_sample.cc
The program includes already the initialization of the lattice and the starting point on a circle. Include in the program (at indicated place) the random walk being done by the particle which starts on the circle. Update the lattice for each random walk step, so that the random walker is visible as a red particle (`lattice[x][y]=2;`) on the blue background with the white seed in the middle. Look at your resulting lattice movie with
`executable | DynamicLattice -nx 100 -ny 100 -matrix`

4. DLA: Distance

4a. In the DLA model you will need to keep track of the distance r of your walker from the midpoint of the lattice. Change your program of 3. such that instead of printing the movie of your lattice, you determine and print for every random walk step two numbers: the step and the distance r.

4b. Now replace the loop of 100 random steps with a while loop `while (walkstop == 0)`. Set `walkstop = 0` before this while loop and set `walkstop = 1`, as soon as $r \geq 2 * R_{\max}$. Initialize $R_{\max} = 3.0$ in the secion of your initialization. Change the radius to `radius = Rmax + 2`. Run your program with `idummy = -6` (to see a few steps).

In-Class Work: Fractal Growth (Newcomers)

3. DLA: Circle and Random Walk Copy into your working directory
~kvollmay/sunhome/classes.dir/capstone_s2008.dir/fractal.dir/DLA3_sample.cc
The program includes already the initialization of the lattice and the starting point on a circle. Include in the program (at indicated place) the random walk being done by the particle which starts on the circle. Update the lattice for each random walk step, so that the random walker is visible as a red particle (`lattice[x][y]=2;`) on the blue background with the white seed in the middle. Look at your resulting lattice movie with
`executable | DynamicLattice -nx 100 -ny 100 -matrix -z 0 2`

4. DLA: Distance (if time)

4a. In the DLA model you will need to keep track of the distance r of your walker from the midpoint of the lattice. Change your program of 3. such that instead of printing the movie of your lattice, you determine and print for every random walk step two numbers: the step and the distance r.

4b. Now replace the loop of 100 random steps with a while loop `while (walkstop == 0)`. Set `walkstop = 0` before this while loop and set `walkstop = 1`, as soon as $r \geq 2 * R_{\max}$. Initialize $R_{\max} = 3.0$ in the secion of your initialization. Change the radius to `radius = Rmax + 2`. Run your program with `idummy = -6` (to see a few steps).

Upcoming Deadlines: (see also our webpage)

- April 4, 5pm: Running Program (readable in share.dir)
- April 10: Results
- April 15: Abstract

Solutions to Random Walk In-Class Work:
~kvollmay/sunhome/classes.dir/capstone_s2008.dir/fractal.dir/rand*.cc

In-Class Work: Fractal Growth (Advanced)

3. DLA: Circle and Random Walk Copy into your working directory
˜kvollmay/sunhome/classes.dir/capstone_s2008.dir/fractal.dir/DLA3_sample.cc
The program includes already the initialization of the lattice and the starting point on a circle. Include in the program (at indicated place) the random walk being done by the particle which starts on the circle. Update the lattice for each random walk step, so that the random walker is visible as a red particle (`lattice[x][y]=2;`) on the blue background with the white seed in the middle. Look at your resulting lattice movie with
`executable | DynamicLattice -nx 100 -ny 100 -matrix -z 0 2`

4. DLA: Distance

4a. In the DLA model you will need to keep track of the distance r of your walker from the midpoint of the lattice. Change your program of 3. such that instead of printing the movie of your lattice, you determine and print for every random walk step two numbers: the step and the distance r.

4b. Now replace the loop of 100 random steps with a while loop `while (walkstop == 0)`. Set `walkstop = 0` before this while loop and set `walkstop = 1`, as soon as $r \geq 2 * R_{\mathrm{max}}$. Initialize $R_{\mathrm{max}} = 3.0$ in the secion of your initialization. Change the radius to `radius = Rmax + 2`. Run your program with `idummy = -6` (to see a few steps).

5. Stick to Cluster (if time)

Next add to your program of 4. that whenever the random walker is next (left, right, top, bottom) to a particle of the cluster then the walker sticks to the cluster. Update the lattice, the number of particles `npart`, the flag for the stopping of the random walk `walkstop` and `Rmax`. Add the necessary lines as indicated in the sample program after the comment "if neighbor stick and stop walk and update Rmax" and use the flow chart from class. Have a look at the resulting movie.

In-Class Work: Fractal Growth (Newcomers)

5. Stick to Cluster

Since in last class most of you got already 4. done (:-), let us today all work on the last two steps of the program. Copy into your working directory
~kvollmay/sunhome/classes.dir/capstone_s2008.dir/fractal.dir/DLA4b.cc
or use your working program from last class for 4b. Next add to the program that whenever the random walker is next (left, right, top, bottom) to a particle of the cluster then the walker sticks to the cluster. Update the lattice, the number of particles `npart`, the flag for the stopping of the random walk `walkstop` and `Rmax`. Add the necessary lines as indicated in the sample program after the comment "if neighbor stick and stop walk and update Rmax" and use the flow chart from class. Have a look at the resulting movie.

6. Finish Program: Loop Over Particles

Now you are ready to finish your DLA program. Add to your program the while loop over particles. Condition for this while loop are both that the wanted number of particles `NPARTMAX` has not yet been reached and that the radius for the start of the random walk fits into the lattice. Use the flow chart of class and add the lines in the program at the with comment indicated places. Use the constants `L=100`, `NPARTMAX=50` and `PAUSENUMBER=0`. When you got your program working, please let me know, so that you can show your movie to the class.

7. DLA Pattern

Next we try to generate a larger cluster to see its fractal structure (and just because it is fun). Print the lattice no longer for every random walk step, but instead print the lattice only once after the while loop over particles. Set `L=500` and `NPARTMAX=3000`. Have fun with the resulting picture. To make a picture which you can put on your wall:

```
DLA7.out > DLApicture.data
lattice2ps -color -nx 500 -ny 500 -matrix < DLApicture.data >
DLApicture.ps
```

The resulting `DLApicture.ps` you can print on a color printer, or to check it on the screen

```
ggv DLApicture.ps
```

In-Class Work: Fractal Growth (Advanced)

5. - 7. See backside.

8. History of DLA Particles

To understand the shape of the DLA clusters, let us now indicate the history of the particles with their color: Print "5" for the first 500 particles, "6" for the next 500 particles, "7" for the following 500 particles etc.. Look at the final result with DynamicLattice.
Hint: Adjust your check if the random walker is next to a particle of the cluster.
If you print only `LMID-100` $\leq$ `i,j` $\leq$ tt LMID+100 you might recognize the resulting picture (:-)

Upcoming Deadlines: (see also our webpage)

- April 15: Results
- April 17: Abstract (no later, so that I can announce your talks)
- April 22: 2nd Paper First Version
- April 22: Symposium Talks WHO?
- April 24: Symposium Talks WHO?
- April 29: Symposium Talks WHO?

2nd Paper: The second paper will be the complete scientific paper about your main project. It will include the sections of your first paper. Use my comments on your revised versions to revise your paper further. In addition you will include now all simulation details, the results section, the summary/future section and the abstract. For further instructions see our webpage and hand-outs in class.

Results: As preparation for your paper, make the necessary figures and tables for your paper. The more you hand in completed figures, the more feedback I will be able to provide to you. On April 15 you should have the complete set of results.

Help/Comments: Please feel free to ask for help any time.

Solutions to Random Walk In-Class Work:
~kvollmay/sunhome/classes.dir/capstone_s2008.dir/fractal.dir/rand*.cc
Solutions to DLA-In-Class Work:
~kvollmay/sunhome/classes.dir/capstone_s2008.dir/fractal.dir/DLA*.cc

In-Class Work: Fractal Growth (Newcomers)

Today we will pick the fruits of our DLA in-class work! Have fun!

6. Finish Program: Loop Over Particles

Use your working program from last class or copy into your working directory
~kvollmay/sunhome/classes.dir/capstone_s2008.dir/fractal.dir/DLA5.cc
This program allows you to print only a middle section of the lattice, i.e. to chop the big blue sea. You can watch the movie with
`DLA5.out | DynamicLattice -nx 40 -ny 40 -matrix`
Or if you want a different size, then adjust LPRINT and the DynamicLattice command. To finish the DLA program add the while loop over particles. Condition for this while loop are both that the wanted number of particles `NPARTMAX` has not yet been reached and that the radius for the start of the random walk fits into the lattice. Use the flow chart of class and add the lines in the program at the with comment indicated places. Use the constants `L=100,` `NPARTMAX=50` and `PAUSENUMBER=0`.

7. DLA Pattern

Next we try to generate a larger cluster to see its fractal structure (and just because it is fun). Print the lattice no longer for every random walk step, but instead print the lattice only once after the while loop over particles. Set `L=500,NPARTMAX=3000` and `LPRINT=250`. Have fun with the resulting picture. To make a picture which you can put on your wall:
`DLA7.out > DLApicture.data`
lattice2ps -color -nx 250 -ny 250 -matrix < DLApicture.data > DLApicture.ps
The resulting `DLApicture.ps` you can print on a color printer, or to check it on the screen
`ggv DLApicture.ps`

8. History of DLA Particles (if time)

To understand the shape of the DLA clusters, let us now indicate the history of the particles with their color: Print "5" for the first 500 particles, "6" for the next 500 particles, "7" for the following 500 particles etc.. Look at the final result with DynamicLattice.
Hint: Adjust your check if the random walker is next to a particle of the cluster.
You probably recognize the resulting picture (:-) When you got your program working, please let me know, so that you can show your movie to the class.

In-Class Work: Fractal Growth (Advanced)

Today we will pick the fruits of our DLA in-class work! Have fun!

6. & 7. See backside.

9. Fractal Dimension of DLA Cluster

In the following you will analyze the pattern of the DLA model. You will determine the fractal dimension of one pattern.

9a. To avoid having to run the DLA program again and again, let us first prepare one pattern, which you then will analyze in 9b. Either use your program of 7. or copy
~kvollmay/sunhome/classes.dir/capstone_s2008.dir/fractal.dir/DLA9_sample.cc
into your working directory and run it with `L=500, NPARTMAX=10000, LPRINT=350` and
`DLA9_sample.out > bigDLAcluster.data`
This prints the final pattern only once at the end of the program into the file "bigDLAcluster.data".

9b. Now write a program which reads in the 350x350 matrix from your file of 9a. To get the fractal dimension d_f we use the following relations.

$$\ln(N) = \ln(c) + d_\text{f} * \ln(b) \quad (2)$$

where N is the number of occupied sites, c is some constant and b is the length of your square for which you count the number of occupied sites. You see that Eq.(2) defines d_f and it tells us that if we plot $\ln(N)$ as a function of $\ln(b)$ then we should get a line with slope d_f. So our task is to get N and b. Add to your program that you count the number of occupied sites N for a lattice of lenght b, where you center your lattice of lenght b around the midpoint of your 350x350 lattice. Loop over the length of your lattice and print out $\text{l}n(N)$ as a function of $\ln(b)$. Let's say you do
`DLA9b.out > lnNoflnb.data`
9c. Next we fit a line to our data from 9b stored in file `lnNoflnb.data`. For this we use gnuplot. So type in the command line "`gnuplot`". Then type "`plot "lnNoflnb.data"`". Define a function $f(x)$ by typing "`f(x) = a*x + b`". Now fit your data within the xrange [2.0,4.5] to a line by typing "`fit [2.0:4.5] f(x) "lnNoflnb.data" via a,b`". The resulting a is the fractal dimension d_f.

Upcoming Deadlines: (see also our webpage)
- April 15: Results
- April 17: Abstract
- April 22: 2nd Paper First Version
- April 22: Symposium Talks: Rob, Sarah, Ze, James
- April 24: Symposium Talks: Austin, Cragin, Beau, Lindsey
- April 29: Symposium Talks: Joshua, Nga, Matt, Jason

Solutions to DLA-In-Class Work:
~kvollmay/sunhome/classes.dir/capstone_s2008.dir/fractal.dir/DLA*.cc

Project III

Today you will work in class on project III in groups. The groups are: Joshua & Nga // Rob & Cragin // Beau & James // Ze & Matt // Austin, Sarah & Jason. Get together with your group partner(s). You will work today on variations and the analysis of the DLA model. Pick one of the projects described below. (Exceptions are projects III.4 & III.5 which would fit perfect with the groups as indicated, since the assigned group members have already worked on fractal dimension analysis.) Our schedule for today is:
9:30 - 10:00 work on project, 10:00 - 10:10 talk preparation, 10:10 - 10:50 talks.
Each group will give a 6 min presentation. You may use the transparencies and pens provided in class. **Have fun!**

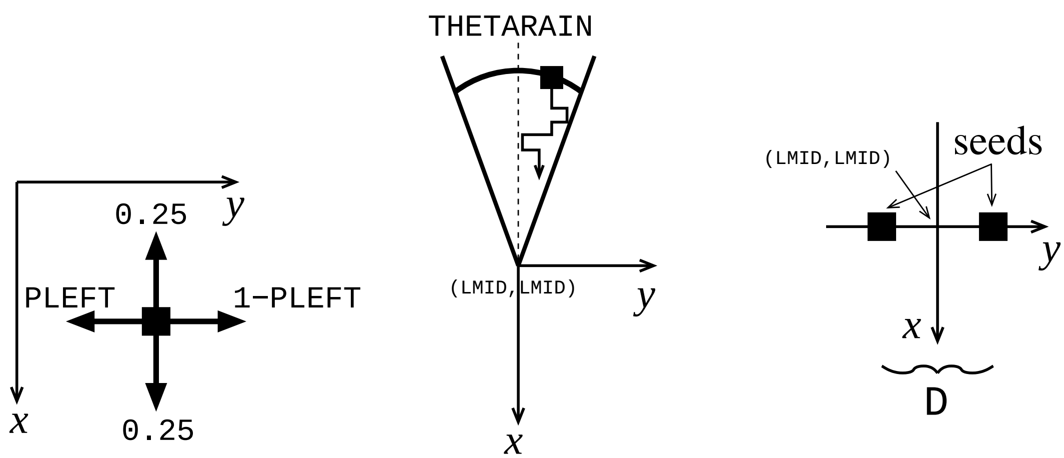

Figure 1 Figure 2 Figure 3

III1. Wind

III1a. Our goal in this project is to model fractal growth with wind. One possible implementation is to change your random walk step. As before walk up or down each with probability 0.25. Instead of having a walk to the right and left of equal probability 0.25, introduce a new constant `PLEFT` which will be set to any number between 0 and 0.5. Let the random walker walk to the left with probability `PLEFT` and to the right with probability `(0.5 - PLEFT)` - see Fig. 1. Use your program from last class or
~kvollmay/sunhome/classes.dir/capstone_s2008.dir/fractal.dir/DLA9_sample.cc
(so `L=500, NPARTMAX = 10000` and the lattice is printed only once at the end for a square lattice of length `LPRINT=350`) and change the program to build in the wind. Have a look at your final fractal pattern.

III1b. Vary `PLEFT` and compare the resulting patterns.

III2. Rain

III2a. Goal of this project is to model fractal growth under the condition that particles are not coming in from all directions. In the original DLA model particles were starting their random walk from a random point on the circle of radius `Rmax + 2` centered about the midpoint of the lattice. Now let the random walk start on only a part of this circle of angle `THETARAIN` as sketched in Fig. 2. You may start with your working program from class or with
~kvollmay/sunhome/classes.dir/capstone_s2008.dir/fractal.dir/DLA9_sample.cc
Look at the resulting pictures of the original DLA model and of the rain featured DLA model.
Hint: Be careful with how `x` and `y` (see Fig. 2) and convert your angle to radians.

III2b. Vary `THETARAIN` and compare the resulting patterns.

III3. Seeds

III3a. In this project you will investigate which influence the initial geometry of your seed(s) has. Use your working DLA program or you may start with
~kvollmay/sunhome/classes.dir/capstone_s2008.dir/fractal.dir/DLA9_sample.cc
Now change the program such that instead of starting with a single seed in the middle of the lattice, you start instead with two seeds centered about the middle of the lattice and a distance `D` apart (see Fig. 3.) Look at the resulting pattern.
Hint: Intialize accordingly not only the lattice but also `Rmax`.

III3b. Vary `D` and compare the resulting patterns.

IV4. Stickyness (James & Beau)

IV4a. In this project you do not always stick a particle when it has a cluster particle as neighbor. Instead you stick only with probability `PSTICK`. Use your working program or you may start with
~kvollmay/sunhome/classes.dir/capstone_s2008.dir/fractal.dir/DLA9_sample.cc
(so 500x500 lattice and `NPARTMAX = 10000` and print only once at the end a square lattice of size `LPRINT=350`). To avoid that a non-sticking particle walks on top of already exitisting cluster particles, do a random walk step only if it would put your random walker onto an empty lattice site. So lines 52-59 are not always done (not for x,y) and similarly do line 50 only if your random walk step is allowed.

IV4b. Now let's compare the **fractal dimension** of the original DLA-model and of the DLA-model with the sticking probability . Use your program and the steps from last class (9b. & 9c.) to determine the fractal dimensions for `PSTICK=1.0` (original) and `PSTICK=0.2`. You may use
~kvollmay/sunhome/classes.dir/capstone_s2008.dir/fractal.dir/DLA9b_screeenin.cc

IV4c. (if time) Determine the fractal dimension for varying `PSTICK`.

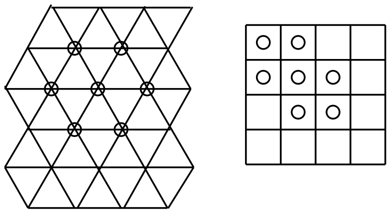

Figure 5

III5. Neighborhood (Snowflake) (Rob and Cragin)

Snwoflakes have a pattern similar to the pattern shown in Fig. 5a. We say that the molecules are on the corners of a triangular lattice. Each molecule is surrounded by 6 neighbors. We have so far always used the square lattice. It turns out that if you look at the triangular lattice of Fig. 5a and tilt your head a little bit to the left, then Fig. 5a looks like a square lattice where the neighbors are right, left, up, down, NW and SE as shown in Fig.5b.

III5a. Use your solution to the in-class work 9a. & 9b. of last class or
~kvollmay/sunhome/classes.dir/capstone_s2008.dir/fractal.dir/DLA9_sample.cc
(so 500x500 lattice and `NPARTMAX = 10000` and print only once at the end a square lattice of size `LPRINT=350`). Change your program such that the sticking occurs according to the neighborhood of Fig. 1b. Run your program and look at the result with DynamicLattice.

III5b. Now let's compare the **fractal dimension** of the original DLA-model and of the DLA-model with the neighborhood of III5a. Use your program and the steps from last class (9b. & 9c.) to determine the fractal dimensions for the two neighborhoods. You may use
~kvollmay/sunhome/classes.dir/capstone_s2008.dir/fractal.dir/DLA9b_screeenin.cc

III5c. (if time) Try other neighborhood definitions and determine the fractal dimension.

Upcoming Deadlines: (see also our webpage)
- April 15: Results (hardcopy of figures and/or tables)
- **April 17: Title & Abstract**
- April 22: 2nd Paper First Version
- April 22: Symposium Talks: Rob, Sarah, Ze, James
- April 24: Symposium Talks: Austin, Cragin, Beau, Lindsey
- April 29: Symposium Talks: Joshua, Nga, Matt, Jason

Solutions to DLA-In-Class Work:
~kvollmay/sunhome/classes.dir/capstone_s2008.dir/fractal.dir/DLA*.cc