In-Class Work: Xfig & Fractal Growth

6. Xfig Intro

I will guide you through the following main commands of `xfig`, which is a drawing tool:

- To get started: Type in the linux environment within a terminal window, so on the command line: `xfig &`
  This will open a new window.
- drawing tools: background grid, circle, line, text, picture, grouping, scaling, copying, editing.
- To save an xfig session use `File` → `SaveAs` and give your xfig-file a name ending with `.fig`. You can get back to this session any time on the command line with `xfig filename.fig &` or within xfig with `File` → `Open`.
- To make an eps-file out of your figure use `File` → `Export`, make sure to choose "EPS (Encapsulated Postscript)" and choose the same filename but with the ending `.eps`. This eps-file can then be included in your latex file for the paper. (Later into the course I will also show you a variation of latex, latex beamer, which we will use to make talk-slides. You will be able to use the same eps-files for the paper and for the talk and therefore your work on the eps-files for your paper will be very handy for your talk preparation.)

7. (optional) Comment for Advanced xfig Users:

In case you would like to use latex commands within xfig use the following steps: First copy
`~kvollmay/share.dir/papertools.dir/xfig2eps`
and
`~kvollmay/share.dir/papertools.dir/xfig2pdf`
then make both executable (these are perl-scripts)
`chmod u+x xfig2*`. These `xfig2*` files will be needed for step (3) below.
Instead of `xfig` use instead
(1) `xfig -specialtext -latexfonts -startlatexFont default`
(2) first save then export to "Combined PS/LaTeX (both parts)."
This creates two files: `filename.pstex` and `filename.pstex_t` . To then make an eps-file (which you can include in your paper) (3a) `xfig2eps filename`
or to make a pdf-file use
(3b)`xfig2pdf filename`

Fractal Growth

1. Fractal Growth: Random Walk in Two Dimensions

1a. For the fractal growth DLA model we will need a random walk in two dimensions. Write a python-program for a random walker on a two dimensional lattice (all four directions being

equally likely), starting at $x = 0$ and $y = 0$ and (print and) look at $x(t)$ and $y(t)$. You may use the solution program `~kvollmay/share.dir/inclass2021.dir/classrndwalk2a.py`.
For looking at $x(t)$ and $y(t)$ in the same figure, you can use the command (assuming that your program is called `classfractal1a.py`)

```
./classfractal1a.py > j; xmgrace -block j -bxy 1:2 -bxy 1:3
```

1b. Movie

Next let's make a movie of your random walk. Define a lattice (`lattice`) of size 30x30 and initialize it for all sites equal to zero. Put your initial walker at site $x = 15$ and $y = 15$. We want to make a movie of the random walker where we mark on the lattice the current random walker site with the lattice value 2 and we mark any previously visited site with 1 (This is just for our fun.). To make a movie we first make an image for every random-walk step. **(So please use only `NSTEPS=50` random walk steps!)** To see how to make these pictures see the example
`~kvollmay/share.dir/pythonsamples.dir/sample_latticemovie.py` Once you have all pictures in `frame*` you can run the movie with

```
animate -delay 30 -pause 5 frame*png
```

2. Fractal Growth: DLA

2a. You may use google to get an idea about the main concepts of the following topics:

- Cellular Automata
- Fractal Growth
- Diffusion-Limited Aggregation

2b. Read in our Gould & Tobochnik textbook about the model we will use, that is about Diffusion-Limited Aggregation (DLA): chapter 13, read bottom of page 529 (skip pages 530 & 531) and read on page 532 the first two sentences of Problem a. (For a link to this book, see our course webpage.) Try to write a flow chart for this DLA model.
Please get me, when you get here. I will give a mini-intro about cellular automata and about fractals.

3. Start Random Walker on Circle

Initialize the lattice with all lattice sites being zero and only in the middle of the lattice is a seed with value one. Use a lattice size `LATSIZE=100`. We want next to implement that a new random walker starts randomly somewhere on a circle with midpoint in the middle of the lattice and with radius `RMAX+2`. For the DLA-program we will use `RMAX=2` but for now use `RMAX=20`. To check that you draw your random walker indeed equally likely on a circle, add to your program that you put 50 initial walkers on their starting point (not yet with random walk steps) and mark for each of these 50 starting points the lattice with the value 2. Make only one image of your lattice after these 50 markings on the lattice. To get a pdf-file of your frame use in the header of program

```
import matplotlib.pyplot as plt
```

and for example

```
plt.imshow(lattice,interpolation='nearest')
plt.savefig('frame3.pdf')
```

Think about how you would program this task. When you have planned your strategy, you may go straight to not writing the program but instead you may copy the solution program `~kvollmay/share.dir/inclass2021.dir/classfractal3.py` into your working directory. Read the program, to convince yourself what it does and run the program. Look at the resulting `frame3.pdf`.

4. DLA: Random Walk Copy `classfractal3.py` into another file, e.g. into `classfractal4.py`. Modify `classfractal4.py`. Take out of this program now the loop over 50 starting points, instead start the random walker at **only one** point on the circle and use `RMAX=2`. Add to your program a loop over `NSTEP=50` random walk steps. To check your program, assign to each site, which is visited by the random walker, the value 2. Print the lattice after this random walk and look at it. Hint: Use your work above from step 1a of the fractal growth section.