First Menu of Projects (for Project II)

A region with four walls. So far, our mechanics example has had very little interesting physics in it. Now we ask you to turn to some more interesting examples. In the homework you wrote a program to depict a situation in which there is a ball bouncing in a square region bounded by four walls. Now for a harder job: do the same thing for an arbitrary quadrilateral.

Physical Question 1. Under what circumstances will the orbit close, i.e. repeat itself?

Physical Question 2. Statistical Mechanics says that in sufficiently complicated situations particles will reach a state of statistical equilibrium in which, all other things being equal, they will spend a time within a given region of the container which is proportional to the area of that region. Since the time spent is proportional to the probability for finding the particle in the region, we have the simple rule equal probabilities for equal areas. Our statement on the conditions required for this rule is vague. However you can be much less vague. Show that this rule is not true in most square or rectangular boxes. For most quadrilateral boxes this rule is true. Construct a numerical demonstration of the plausibility of the equal areas rule in some quite unsymmetrical quadrilateral.

Dynamics in the Stadium. A 'stadium' is a couple of semicircles stuck on the ends of a rectangle.

First consider a situation in which there is a ball bouncing within a circular region. Under what circumstances will the

* * *

orbit close, i.e. repeat itself? Is this circumstance likely or unlikely? Then consider the motion of a point particle which bounces around a stadium. Under what conditions does this orbit close?

Someone suggests that in a stadium, for ‘most’ starting conditions the probability that the particle will be moving with an angle q to the x-axis is independent of q. Pick a particular stadium and choose some starting condition ‘at random’ and see whether you can offer evidence for or against this hypothesis.

In doing this you will have to more sharply define the hypothesis mentioned above. Specifically, you will have to choose between two alternatives: Does the particle have equal numbers of trajectories at each angle or does it spend an equal time traveling at each angle? Note: the Java Math method Math.random() generates a double value from 0 up to (but not including) 1.

Dynamics in the Triangle. Take a triangle with sides which have lengths L_1, L_2, and L_3. There is a ball bouncing in the triangle. Can you find the law which says, for most starting conditions of the ball, what is the relative probability that the ball will hit each of the sides?

Hint 1: The law is simple.

Hint 2: Start with a case in which one length is very much smaller than the others.

Redo Problem 3.1 for the complex roots of the function $f(z) = z^3 - 7z + 6$. The Newton-Raphson method works just the same way for complex numbers as reals. Try to show how the root found depends on the starting value of z. Generate some graphical representation of your answer. (You can see some of

* * *

the answer on pages 116 and 117 of a book by Peitgen and Richter.[1])

Using a turtle to draw fractals. This project consists of two parts. The first is to create a "turtle" graphics system (much like LOGO). The turtle graphics system has a turtle at some position pointing in a certain direction. The turtle can move forward, leaving a trail of ink behind (drawing a line), and it can rotate. It can also turn the ink on and off. So your class has to support the following commands:

```
public class Turtle {
   public Turtle(Graphics g, int height, int width, int xpos, int ypos, int angle);
   // constructor.  Draws into the graphics context g, which has height height and
   // width width.  Starts turtle at initial position (xpos, ypos) and angle angle.

           public void right(int angle);   // rotates turtle right by angle degrees
           public void left(int angle);    //  rotates turtle left by angle degrees
           public void pendown();          //  puts turtle's pen down (turns ink on)
           public void penup();   //   puts the turtle's pen up (turns ink off)
           public void forward(int distance);   //  moves forward distance steps.
                   // if pen is up, doesn't draw anything; if pen is down, draws line
           public void forward(double distance); // optional.  Use double precision
                   // variables for extra accuracy
}
```

(See also problems 5.21 and 9.29 in Deitel and Deitel.)

The second part of the project is to use your turtle to draw fractals. In the appendix of Chapter 4, we describe turtle commands needed to draw one particular fractal, the Koch curve. Have your turtle make the Koch curve and/or other fractals of your choosing.

[1] H-O. Peitgen and P.H. Richter The Beauty of Fractals Springer-Verlag Berlin 1986.

***

Julia sets. In this project you will investigate the map

$$f(z) = z^2 + c,$$

where z and c are both complex. We are again interested in the set of z whose orbits are bounded, meaning that $|f^n(z)| < \infty$ for all n. See what this set looks like when the parameter value c = -0.5 + 0.5i. What happens when the parameter c is changed? You may find chapter 13 of Peitgen, Jurgens, Saupe, Chaos and Fractals: New Frontiers of Science, Springer-Verlag (1992) useful for this project.

Diffusion Limited Aggregation (DLA). This project explores the use of random walks to model an interesting growth process of dendritic structures made up of particles that stick together when they come in contact. The particles are assumed to diffuse slowly through space until they come in contact with another particle, at which time they stick. The computer implementation is simple in concept: On a lattice (i.e. an array of points) put a particle at the origin (the "seed"). Then start another particle from far away, and allow it to execute a random walk (with step size equal to the lattice spacing). After a time, it will land on one of the points adjacent to the particle at the origin. Stop the particle there, and begin another one from far away etc. The structure that develops has a beautiful, tree-like, fractal shape. Unfortunately, it grows very slowly. This is a project that you will want to run as long as possible. Compute the fractal dimension of your clusters, defined by the relation N R^d, where N is the number of particles in the cluster, R is the "radius" of the cluster (typically the radius of gyration), and d is the fractal dimension. Grow several large aggregates to get an idea of the accuracy of your estimate for d.

A river network model. Chapter 19 of Gould and Tobochnik (2nd edition) discusses one model of river

* * *

networks (from R.L. Leheny, Phys. Rev. E 52, 5610 (1995)), in which a rectangular lattice of points describes an eroding terrain with the height of the land, h(x,y), specified at each point. The simulation begins with the landscape as a featureless incline: h(x,y) = Iy. Then the following rules are implemented:

1) Precipitation lands at a random site on the lattice.

2) Water flows from this site to one of the four nearest neighbors with a probability proportional to $e^{E\Delta h}$, where Δh is the height difference between the site and the neighbor, and E is a parameter of the model. If $\Delta h<0$, this probability is set equal to zero.

3) Step 2) is repeated until the water reaches the bottom of the lattice, y=0.

4) Each point that has been visited by the flowing water has its height reduced by a constant amount D. This process represents erosion.

5) Any site at which the height difference Δh with a neighbor exceeds a threshold M is reduced in height by an amount $\Delta h/S$, where S is another parameter in the model.

Write a program that implements this model. You can get an idea of suitable parameters to use from Leheny's paper. The resulting river network is defined as follows: every lattice point receives one unit of precipitation which traces a path of steepest descent, without eroding the terrain, until it reaches the lattice edge, y=0; the river network is defined as all points through which at least R units flow. Analyze the network that is generated at different times. Does the river network appear to be fractal? How does evolving the model for longer times affect the network's properties?

* * *