Name: ____________________, ____________________

PHYSICS 401 : SPRING SEMESTER 2018

Project #6: Fractals

Reference Reading: Sections 6.3, 6.8 13.1-2, 13.4

Downloads: Downloaded Chapter 6 from
http://www.opensourcephysics.org/items/detail.cfm?ID=7375 .
Optinal: You may wish to downloaded Eclipse from http://www.eclipse.org/downloads/

1. Do problem 6.5 on P.153, parts (a)-(c). You can use anything at your disposal, gnuplot, your own program, or any of the softwares in `workspace_compadre`.

2. Do problem 6.17 on P.173, parts (a)-(e). Just run `PoincareApp`. Remember to hit the `clear` button often to remove transient behaviors.

3. The fractal fern demonstrated in class was created by using affine transformations of the form

$$x_{n+1} = a(i)x_n + b(i)y_n + e(i),$$

$$y_{n+1} = c(i)x_n + d(i)y_n + f(i),$$

where the coefficients for the four transformations used are

i	$a(i)$	$b(i)$	$e(i)$	$c(i)$	$d(i)$	$f(i)$	$p(i)$
1	0.80	0.04	0.00	-0.04	0.80	0.20	0.75
2	0.23	-0.23	0.00	0.23	0.23	0.15	0.10
3	0.23	0.23	0.00	-0.23	0.23	0.18	0.10
4	0.00	0.00	0.00	0.00	0.25	0.00	0.05

The last column gives the probability of using that transformation, *i.e.* three quarters of the time we'll use transformation 1, a tenth of the time we'll use transformation 2 and 3, and one twentieth of the time we'll use transformation 4. Write a program to generate this attractor. Program notes:

a) The array notation is used to convey compactly the numerical coefficients. You may but do not need to use arrays to do this assignment. Simply write out each transformation using the above numerical coefficients. Denote your program as `FractalApp.java`.

b) Starting with any initial values, say $x_0 = 0.5$ and $y_0 = 0.5$, generate a succession of points (x_n, y_n) according the above transformations.

c) To use transformation i with probability $p(i)$, define the *accumulated* probability $q(i) = \sum_{j=1}^{i} p(j)$ (*i.e.* q_1=0.75, q_2=0.85, q_3=0.95, q_4=1.00), and do the following "dice rolling" to select each transformation:

```
rdn=Math.random(); \\ generate a random number uniformly in [0,1)
if(rdn<0.75) {
    do transformation 1 }
else if(rdn<0.85){
    do transformation 2 }
else if(rdn<0.95){
    do transformation 3 }
else {
    do transformation 4 }
```

Explain why this then samples each transformation with the required probability.

d) Discard the first 100 iterations and plot the remaining fixed points as in the logistic map case. (You probably need 10,000 to 40,000 points.) When you plot this out, make sure that the aspect ratio is correct in producing a nice looking, rather than a distorted looking, fern. (Continue to the next page.)

e) The rotation angles in transformation 2 and 3 are exactly 45°. If you narrow this angle, you can change the fern to a corn stalk or a tall grass. If you increase the angle in transformation 1, the fern will wilt. If you assign a probability of zero, that transformation will not be used. Play around with these and other parameters in the above tranformations and hand in one additional image that you have devised on your own.

4. (Optional) You might want to try this too.

i	$a(i)$	$b(i)$	$e(i)$	$c(i)$	$d(i)$	$f(i)$	$p(i)$
1	0.25	0.00	0.00	0.00	0.50	0.30	0.10
2	0.50	-0.30	0.00	0.30	0.50	0.50	0.40
3	0.30	0.40	0.00	-0.40	0.30	0.50	0.40
4	0.00	0.00	0.00	0.00	0.45	0.00	0.10

What happens if you set p(1)=0.0?