Student name: ***Enter your name here***

Combined Computer Lab and Homework Assignment 3

Logistic Map

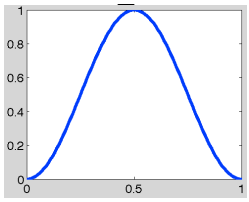
(1) Please download two files *sin2.m* and *run_sin2.m* from bCourses. The first file

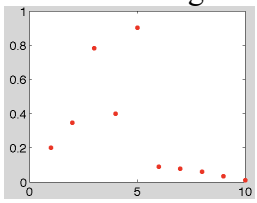
```
function y = sin2(x)
xx = x .* pi
y = sin(xx) .* sin(xx);
```

is a Matlab function that is *called* from the second file. This function works like any other Matlab function (e.g. sin(x)) but you can decide what it does based on the input, x. The second file says this:

```
x = 0:0.01:1;
y = sin2(x);              % NEW: this is a call to your own function

axes('FontSize',20);
plot(x,y,'LineWidth',5); % plot 1
pause                     % waits until you press RETURN

x = 0.2;  % starting point for iteration
for i = 1:10
    xx(i) = i;
    yy(i) = x;
    x = sin2(x) % assign the next value in the iteration to 'x'
end

plot(xx,yy,'r.','Markersize',20); % plot 2
```

Read through both files. Run *run_sin2.m* and reproduce the following images:



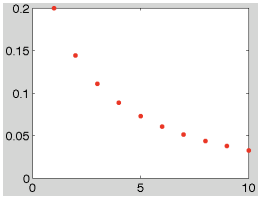


Now remove the `pause` command, insert `"figure, "` before each plot command and run it.

(2) Now save the function file *sin2.m* under the new name *logistic_map.m*, introduce a second function argument *r* such that

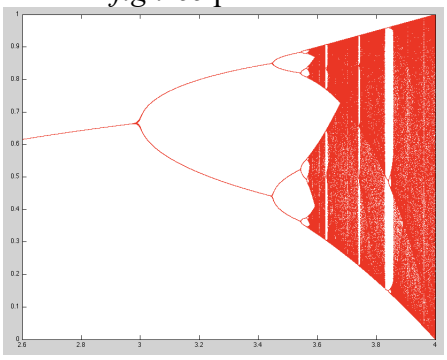
```
function y = logistic_map(x,r)
```

and replace `sin²(pi*x)` by `r.*x.*(1-x)`. Then change *run_sin2.m* to *run_logistic_map.m*, delete the first section (from "x = 0:0.01:1" to "pause") and replace all function calls to *sin2(x)* to *logistic_map(x,r)*. Keep r=0.9 fixed for the moment. Run the file run_logistic_map.m and check that you see:



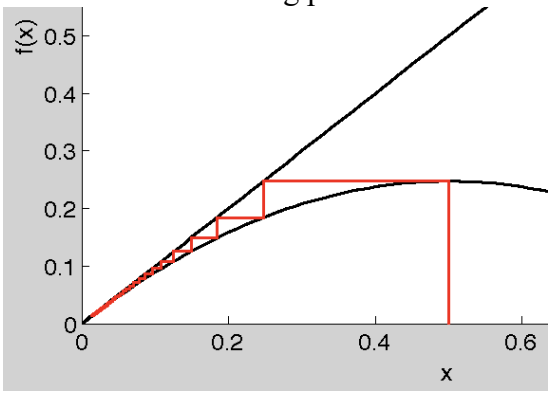
(3) Run for 100 instead of only 10 iterations. **Enter 6 plots here** for the following *r* values: 2.5, 3.3, 3.5, 3.56, 3.80, and 3.83. Make sure you give a title to each graph that indicates its *r* value. *(Instead of plotting 6 individual figures, it is possible to plot all 6 together using the command “subplot”. Look it up in Matlab help (or online) if you are interested. This is optional.)*

(4) Now we want reproduce the famous *fig tree* plot that was discussed in the lecture:



Please use a new script file! Enter an additional loop over *r* values ranging from 2.6 to 4.0. Inside this loop, start a new series of 300 iterations starting from x=0.5 for each *r* value. For the first 200 iterations, do not plot anything since we want to ignore the initial points where the system has not yet settled down. For the following 100 iterations, plot x_i versus *r*. This means x_i will be the ordinate (Y axis) values, *r* specifies the value on the abscissa (X axis). **Submit your Matlab code and fig tree plot in a high resolution with 500x500 pixels or more.**

(5) Finally, we want to reproduce the cob-webbing pictures such as



Go back to the file from section 3. Make sure a plot that contains *f(x)=x* and *f(x)=r*x*(1-x)* before you start adding any red lines. Issue the command “`hold on`” so that additional plot commands will be *added* to the graph rather erase it. Have a careful look at the points the red line. Before trying to code anything please fill in the x and y coordinates of the first coupe of point on the red line:

	X coordinate	Y coordinate
First point	x_0=0.5	0
Second point	…	…
Third point	…	…
Fourth point	…	…
Fifth point	…	…
Sixth point	…	…

There are different ways to add the red line into an existing diagram. The simplest approach is to add every line segment with separate plot command. Normally we give many X and Y values to the plot command at once but it works just as well if we just provide two X and two Y coordinates:

```
plot([xB xE],[yB yE],'r-);
```

Please **enter your 6 cob-webbing pictures** for r=2.5, 3.3, 3.5, 3.56, 3.80, and 3.83.