6.094 Introduction to MATLAB®
January (IAP) 2009

Homework #2: Visualization and Programming

What to turn in: Please turn in a document (preferably Word or PDF) of your work including code and plots. Be sure to suppress intermediate output. Assume you should provide final output unless told otherwise. Note that the last two problems are optional!

Exercise 1. Logical Masks

1. Do Problem 8 in Chapter 4 on page 242 of Palm. Do not use a loop. Do it in two different ways, in the first use a relational operator as a mask and the `sum` function. In the second, use the `find` function.
2. Do Problem 9 in Chapter 4 on page 242 of Palm. Do it in two different ways too, using a logical mask and using the `find` function.

Exercise 2. Conditional Operators and Functions

Do Problem 17, part(b) in Chapter 4 on page 244. Remember to thoroughly test your function.

Exercise 3. To Loop or not to Loop

Do Problem 20 in Chapter 4 on page 246 of Palm. Evaluate to a precision 1/1000 second. Use the `tic/toc` commands to benchmark how long each of the two methods take. Which one is faster? Be sure to suppress output until after you write "toc." Printing or plotting to the screen takes the most time of all!

Exercise 4. Brute-force Optimization

Do Problem 25 in Chapter 4 on page 248-249.

Exercise 5. One-Dimensional Plotting

Do Problem 18 part (c) in Chapter 5 on page 344 of Palm. Overlay all of the trajectories in different colors and/or linestyles on the same plot. The plot should be clearly labelled and have a legend. We should be able to see the apex of the trajectory, in addition to the point it hits the ground. Further, assuming the $y = 0$ is the ground, we should not be able to see the trajectory for negative y.

Exercise 6. Parametric Curves

1. The limaçon is a polar curve of the form

$$r = b + a\cos\theta \tag{1}$$

   If $b \geq 2a$, the limaçon is convex. If $2a > b > a$, the limaçon is dimpled. If $b = a$, the limaçon degenerates to a cardioid. If $b < a$, the limaçon has an inner loop. Write a script that plots the limaçon for the four different regimes, clearly labelling which values of a and b were used. You will find the function `polar` useful.

2. Lissajous curves are a family of curves described by the parametric equations,

$$x(t) = a\sin(\omega t + \delta) \tag{2}$$
$$y(t) = b\sin t \tag{3}$$

Fix $a = b = 1$ and $\delta = 0$. Plot the curve for various values of ω to help answer the questions below. You may want to write a script that will generate your plots. When increasing the range of t, make sure you extend it enough to give the function a chance to close on itself.

(a) Pick ω as an integer and plot the Lissajous curve as t varies over the range $0 < t \leq 2\pi$. How does the value of ω control what the curve looks like? Increase the range of t, what happens (e.g. $0 < t \leq 100\pi$) ?

(b) Now pick ω as a rational number (e.g. $\omega = 2/3$) and plot the Lissajous curve as t varies over the range $0 < t \leq 2\pi$. Increase the range of t, what happens? Does the curve eventually close on itself, if we make t vary over a large enough range?

(c) Now pick ω as an irrational number (e.g. $\omega = e$, π, $\sqrt{3}$) and plot the Lissajous curve as t varies over the range $0 < t \leq 2\pi$. Increase the range of t, what happens? Does the curve eventually close on itself, if we make t vary over a large enough range?

Exercise 7. 3-D Curve and Surface Plotting

1. Make a surface plot of the function $f(x) = \sqrt{2 - x^2 - y^2}$ on the range $[-1, 1]$ in both x and y. Sample this range finely enough to get a smooth shape, and use flat shading with a hot colormap.

2. Make a surface plot of the function $f(x) = xy\sin\left(\frac{x}{y}\right)\cos\left(\frac{y}{x}\right)$ on the range $[-10, 10]$ in both x and y. Sample this range finely (0.1 should be good enough) and use interpolated shading. Change the colormap, add a colorbar, and rotate the plot to a new orientation using MATLAB's plot GUI.

Exercise 8. Vector Field Visualization (Optional)

The electric potential vield V at a point, due to four charged particled is given by,

$$V = \frac{1}{4\pi\epsilon_0}\left(\frac{q_1}{r_1} + \frac{q_2}{r_2} + \frac{q_3}{r_3} + \frac{q_3}{r_3}\right) \tag{4}$$

where q_i is the charge of particle i in Coulombs (C), and r_i are the distances of the charges form the point (in meters), and ϵ_0 is the permittivity of free space, whose value is

$$\epsilon_0 = 8.854 \times 10^{-12}\ \mathrm{C^2/N \cdot m^2} \tag{5}$$

Suppose the charges are $q_1 = 2 \times 10^{-10}$ C, $q_2 = -4 \times 10^{-10}$ C, $q_3 = 6 \times 10^{-10}$ C, and $q_4 = -8 \times 10^{-10}$ C. Their respective locations in the xy plane are $(.3, 0), (-.3, 0), (0, .3)$, and $(0, -.3)$.

1. Plot the electric potential field on a 3D surface plot with V plotted on the z-axis ove the ranges $-0.25 \leq x \leq 0.25$ and $-0.25 \leq y \leq 0.25$. Create the plot using the `surf` function.

2. Given the electric potential, we can compute the electric field using the expression,

$$\vec{\mathbf{E}} = -\nabla V \tag{6}$$

where the ∇ operator represents the gradient operation. Find the E-field on a grid of points over the same range as in (a). Approximate the gradient numerically using the `grad` function. In MATLAB 2008 `grad` has been superceded by `gradient`. Note that you will have two matrices to contain the E-field, one that contains $\vec{\mathbf{E}}_x$, the field in the x-direction, and another $\vec{\mathbf{E}}_y$, the field in the y-direction.

3. Plot the vector field using `quiver`.
4. Plot the E-field lies using `streamslice`.
5. Overlay 100 isocontours of the electric potential V on top of the plot of the streamlines. Use `contour`. What do you notice about the potential isocontours and the steamlines from the previous part?

Exercise 9. The Logistic Map (Optional)

In this problem you will explore the behavior of a famous iterated function called the logistic map. It will introduce you to the art of numerical simulation, for which MATLAB is known for. In addition, you will see how the same data can be visualized in many different ways using MATLAB. The following description is adapted from Wikipedia at `http://en.wikipedia.org/wiki/Logistic_map`.

The logistic map is a polynomial mapping, often cited as an archetypal example of how complex, chaotic behaviour can arise from very simple non-linear dynamical equations. The map was popularized in a seminal 1976 paper by the biologist Robert May. The logistic map models the population of a species in the presence of limiting factors such as food supply. It has two causal effects:

- reproduction means the population will increase at a rate proportional to the current population
- starvation means the population will decrease at a rate proportional to the value obtained by taking the theoretical "carrying capacity" of the environment less the current population.

Mathematically this can be written as,

$$x_{n+1} = rx_n(1 - x_n) \tag{7}$$

where:

- x_n is a number between zero and one, and represents the population at year n, and hence x_0 represents the initial population (at year 0)
- r is a positive number, and represents a combined rate for reproduction and starvation.

1. Write a function `xvec = logistic(x0,r,N)` that takes in three parameters, `x0`, the initial condition of the population, `r`, the combined growth rate, and `N`, the number of years for which to evaluate the population size. The function should return a vector, `xvec`, of length N that contains the evolution of the population over N years, i.e. the first value should be x_0 and last value should be x_{N-1}. This is called the 'orbit' of x_0.
2. Plot the orbit versus year for various values of $0 < r < 4$. You will find that there are four different steady-state regimes. The first, for $0 < r < 1$, the orbit, no matter what the initial condition, dies down to zero. The second, for $1 < r < 3$, the orbit settles down to a single steady-state value. The third, for $3 < r < 3.57$, the orbit settles down to a periodic steady-state (of period 2, 4, 8, as r increases). And the last, for $3.57 < r < 4$, the orbit bounces around seemingly randomly among various values, never settling down into a pattern. This type of behavior is called "chaos". For $r > 4$, the population diverges. Find one example of each of the four regimes and plot them on the same figure using `subplot`.
3. Write a function `firstreturn(xvec, r)` that takes in two parameters `xvec`, an orbit, and `r`, the value of r that generated it. The function should make a two-dimensional scatter-plot of x_n versus x_{n+1} the form in Figure 1.

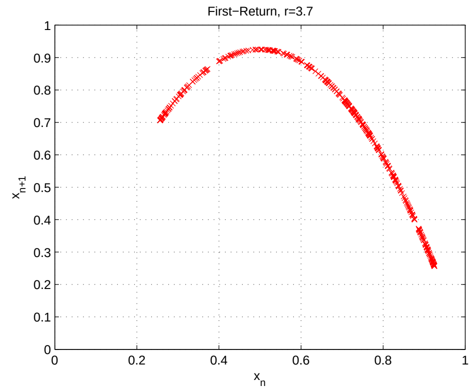

Figure 1: Example First-Return Map

4. Plot the first-return map for the various examples you tried in part (b). In addition, plot the first-return map for the chaotic orbit generated by $r = 3.99$. Note that when plotting the first-return map, you usually do not want to include the 'transient' values near the beginning of an orbit, before it settles into a steady-state pattern. Also, note how that, even though the orbit seems to be completely random, the first-return map shows a quadratic structure to the iteration.

5. Write a script `bifurcation.m` that makes a plot of the steady-state of orbits versus r. You can do this by iterating an orbit for $N = 250$ iteration and then plotting the last $M = 30$ values (which we consider to be steady-state) versus the value of r used. This is called a bifurcation diagram. The resulting plot will look something like Figure 5, where we have plotted the bifurcation diagram of the logistic map for $0 < r < 3.57$.

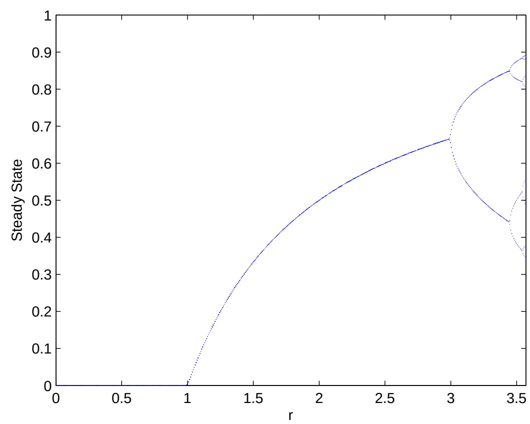

Figure 2: Example of a Bifurcation Diagram

Plot the bifurcation diagram of the logistic map for 1000 equally spaced points in the range $2.4 < r < 4.0$. It will show the different regimes for the logistic map, i.e. you should be able to clearly see how the regimes change from one steady-state to periodic steady-states that double in period and finally to a chaotic pattern.