

## Project 1: Getting to know Matlab

In the `coffee.m` example, we created an array of values  $x(n)$  satisfying

$$x(n+1) = x(n) - \frac{1}{N} \cdot x(n). \quad (1)$$

In this project we will study the <sup>1</sup>equations

$$x(n+1) = x(n)^2 - y(n)^2 + c, \quad (2)$$

$$y(n+1) = 2x(n)y(n) + d. \quad (3)$$

Unless otherwise specified, let  $c = -0.8$  and  $d = 0.156$ .

- a. Modify (or write your own) code to solve the above equations.
- b. For specific starting point `x(1)` and `y(1)` (let's say, `x(1)=0.1`, `y(1)=0.1`), plot the first 22 values of `x(n)` versus `n`.
- c. For specific starting point, plot first 22 values `y(n)` versus `x(n)`.
- d. Write code to create 100 numbers, uniformly randomly selected from the interval  $(-2,2)$ .
- e. For `NStartingPoints=100` uniformly random in  $(-2,2)$ , plot `x(1)` versus `y(1)`. Note this means there are 200 random numbers. This should be a uniform random spread of dots.
- f. For `NStartingPoints=100` uniformly random in  $(-2,2)$ , check if each `x(1),y(1)` was outside the box  $(-2,2)$ . If so, plot it in red. If not, plot it in blue.
- g. Do this for `NStartingPoints=1e5`.
- h. Change parameter value  $c$  and  $d$  (to whatever you want) and repeat.
- i. Bonus Part: Make a version that records what `n` each initial `x,y` leaves the  $(-2,2)$  box. Call this `n_at_exit`. Then, when you plot the points that exist, color the points by the `n_at_exit`.

---

<sup>1</sup>first studied by Gaston Julia, not to be confused with JuliaLang.

---