Assignment 2

1 Programming and Python libraries

(*50 Points*) **Orbits**. Many problems in science and mathematics involve iteration. In dynamics, the process that is repeated is the application of a function. To iterate a function means to evaluate the function over and over, using the output of the previous application as the input for the next. Mathematically, this is the process of repeatedly composing the function with itself. Given $x_0 \in \mathbb{R}$ we define the orbit of x_0 under F to be the sequence of points $x_0, x_1 = F(x_0), \ldots, x_n = F^n(x_0)$[1]. The point x_0 is called the *seed* of the orbit.

With this in mind, let us play with the following function:

$$F(x) = \begin{cases} 2x & 0 \leq x < \frac{1}{2} \\ 2x - 1 & \frac{1}{2} \leq x < 1 \end{cases}.$$

1. Choose 5 different initial seeds that should be in the interval $[0, 1)$ and for each one compute the first 100 points on the corresponding orbit. Record the results by listing the initial seed together with what happened to the orbit.

   (a) (*20 Points*) Plot the results of two cases in a plane x_i vs x_{i+1}.

   (b) (*10 Points*) Do they have a visible pattern? Do all (or almost all) orbits behave in the same way?

   (c) (*20 Points*) For the same two discussed cases, instead of treating the seeds a floating point numbers, use *SymPy* to do the exact rational arithmetic. Do the same for the seed $x_0 = \frac{1}{9}$. What do you find for these three cases? Does the computer lie?

[1] Note that $F^n(x)$ does not mean raise $F(x)$ to the nth power (an operation we will never use). It is the nth iterate of F evaluated at x.