**Scientific Programming in Python**
**Worksheet**
**29 September 2009**

# The Malthus Model

According to the Malthusian growth model the size of a population (of people, elephants, bacteria, etc.) grows with a constant factor every year. For example, the average growth rate of the number of humans on the planet is roughly 1.17% per year. This means that if there were $6.71\times10^9$ humans at the beginning of 2009, there will be

- $(6.71 \times 10^9) \times (1.0117)^1 \approx 6.79 \times 10^9$ at the beginning of 2010,
- $(6.71 \times 10^9) \times (1.0117)^2 \approx 6.87 \times 10^9$ at the beginning of 2011, and
- $(6.71 \times 10^9) \times (1.0117)^{11} \approx 7.63 \times 10^9$ at the beginning of 2020.

The discrete update equation for this model is

$$\begin{aligned} p_{t+1} &= p_t + g\,p_t \\ &= (1+g)p_t \end{aligned} \tag{1}$$

where $p_t$ is the size of the population at time $t$ and $g$ is the growth rate of the population. As an exercise you can show that the closed form solution of this model is

$$p_t = (1+g)^t p_0 \tag{2}$$

where $p_0$ is the initial population size.

**Task 1** Generate a plot like Figure 1(a) to show what population growth looks like according to the Malthus model. You can use either (1) or (2) to generate data for the plot. You should try different values for $g$ where some are positive and some are negative. What happens if $g = 0$? This is unfortunately not a very realistic model of populations. Note that in this model if $g > 0$, $p_t \to \infty$ as $t \to \infty$, and if $g < 0$, $p_t \to 0$ as $t \to \infty$. So the population either grows infinitely large or drops to 0 depending on the growth rate. Also, comparing the exponential curve of Figure 1 to actual data in Figure 1(b) we see that we do not yet have a good model.

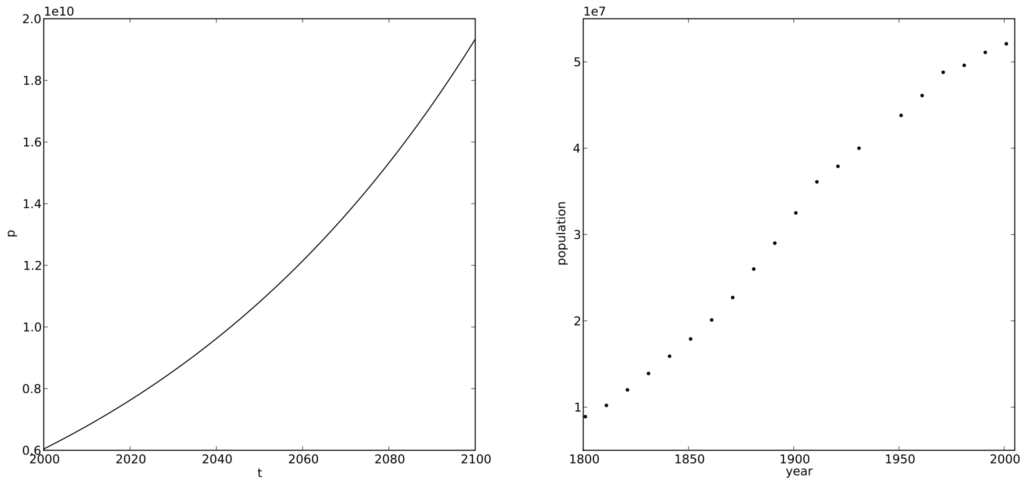
# The Logistic Model

To remedy this situation we need to take into account that any environment has a limited food source for its populations. Populations compete for the available food. If the population is too large and the available food insufficient to support it, the population should shrink. If there is more than enough food, the population should grow as in the Malthus model. s A simple way to incorporate all of this into a new model is to multiply the growth rate, $g$, with a factor that drops to 0 as the population becomes too large. Our new update equation becomes

$$p_{t+1} = p_t + g\left(1 - \frac{p_t}{c}\right)p_t \tag{3}$$

where $p_t$ and $g$ are defined as before and $c$ is the *carrying capacity* of the environment. The carrying capacity is the maximum population that the environment can support. Note that as the population grows towards the carrying capacity, the growth rate, $g\left(1 - \frac{p_t}{c}\right)$, drops to 0.

$$p_t \to c \quad \Rightarrow \quad g\left(1 - \frac{p_t}{c}\right) \to 0. \tag{4}$$




(a) (b)

Figure 1: (a) The growth of a population according to the Malthus model with $p_{2009} = 6.71 \times 10^9$ and $g = 1.17\%$. (b) The population of England from 1801 to 2001.

To explore the update equation mathematically, we rescale

$$\begin{aligned} r &= 1 + g \\ x_t &= \frac{g}{c(1+g)} p_t \end{aligned}$$

so that (3) becomes

$$x_{t+1} = r x_t (1 - x_t) \tag{5}$$

and look at what happens for different values of $r$ in the range $[0, 4]$.

**Task 2** Write a Python program that plots $x_t$ against $t$ for different values of $r$. Try the following values

- $r = 0.8$
- $r = 2$
- $r = 3.3$
- $r = 3.5644$
- $r = 3.569947$

You should see that $x_t$

- converges to 0 for $r < 1$,
- converges to a single value for $1 < r < 3$,
- then converges to cycles of various lengths for larger values of $r$ (for example, there is an 8-cycle at $r = 3.5644$,
- shows no convergence as $r$ approaches a value around 3.569947, and the behaviour becomes chaotic.

Also, confirm that the convergence behaviour does not depend on the initial population size, $x_0$.

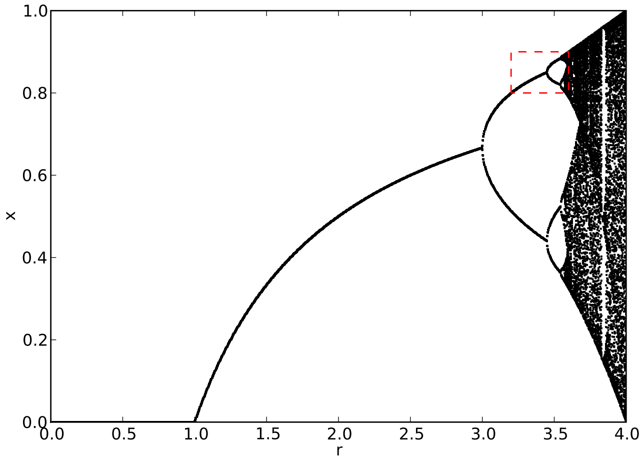
**Task 3** Next, we are going to visualise the stabilities of the logistic map for different values of $r$. You should already have seen that the logistic map converges to different values and also different cycle lengths as $r$ is varied.

One problem that we need to overcome is that we want to plot all of the values in any limit cycle without knowing what the length of the cycle is *a priori*. To address this, we are going to cheat a bit, but not so much that it will affect the visualisation.

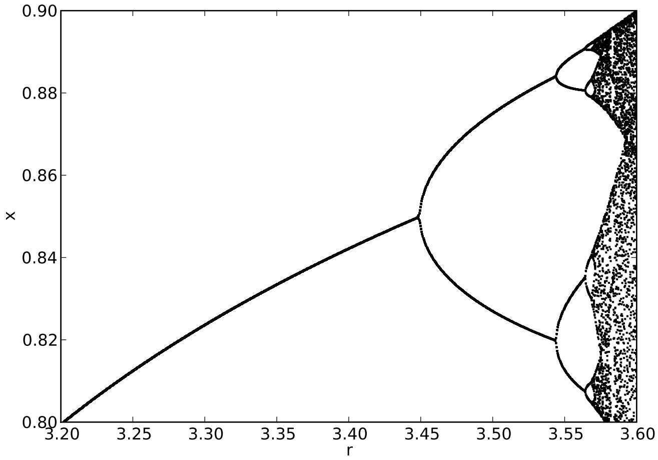
By plotting the graphs of the previous task you can see that the logistic map typically converges after a small number of iterations. We take advantage of this by making the assumption that the system always converges after $N = 1000$ steps. The specific value for $N$ is somewhat arbitrary, so feel free to make it larger.

After $N$ iterations, we plot the next $n = 100$ values of $x$ against a single value of $r$. If the system converged to one value, all of these points will in fact be plotted at the same coordinates and will appear as one point. If the system settled into a 2-cycle, the $n$ points will appear as 2 points, etc. Thus plotting the last $n$ points has the effect of drawing the correct image for any cycles with length at most $n$—again, feel free to increase the value of $n$ beyond the one suggested here. If you plot the points for different values of $r$ in this way, you should get the Feigenbaum tree as in Figure 2.

The second plot shows what happens when you zoom into the boxed area on the first plot. Note how similar the second is to the first even though it is simply a small part of the original plot. This is known as self-similarity and is a typical attribute of chaotic and fractal systems.




(a)




(b)

Figure 2: The Feigenbaum tree for (a) $r \in [0, 4]$ and (b) zoomed in on $r \in [3.2, 3.6]$.