Assignment #6:
Euler's Method and the Logistic Model

Timmy Nguyen

March 13, 2017

1 A Brief Introduction

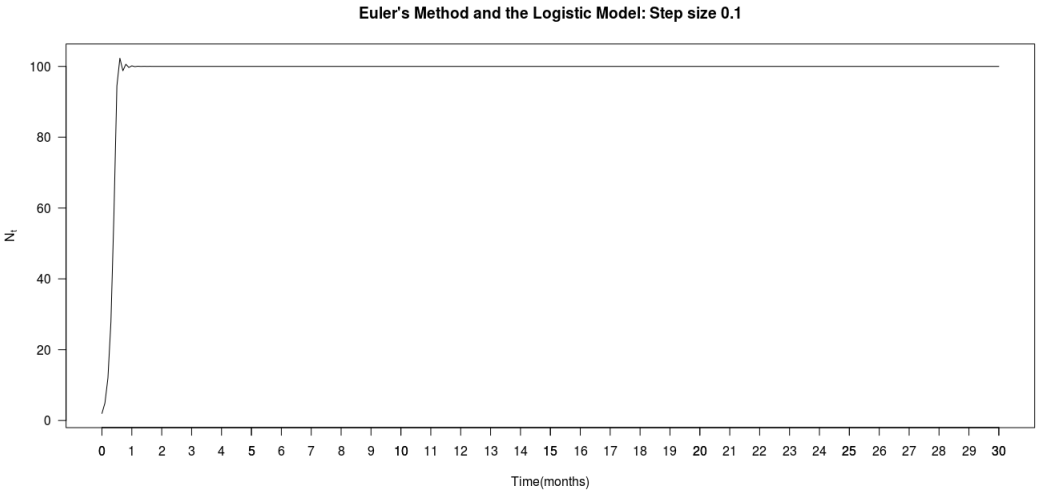
In this assignment, we modified our previous logistic growth model using differential equation. Where the DE model is represented using:

$$\frac{d_N}{d_t} = r * N * (1 - \frac{N}{K})$$

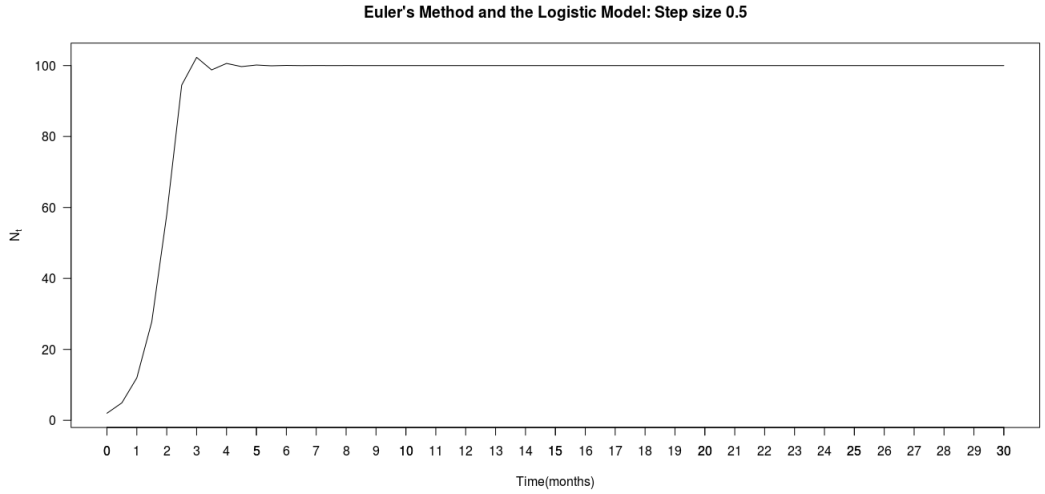
Here the rate of growth, r was 1.5 per month and the carrying capacity, K, was set to 100.

2 Results

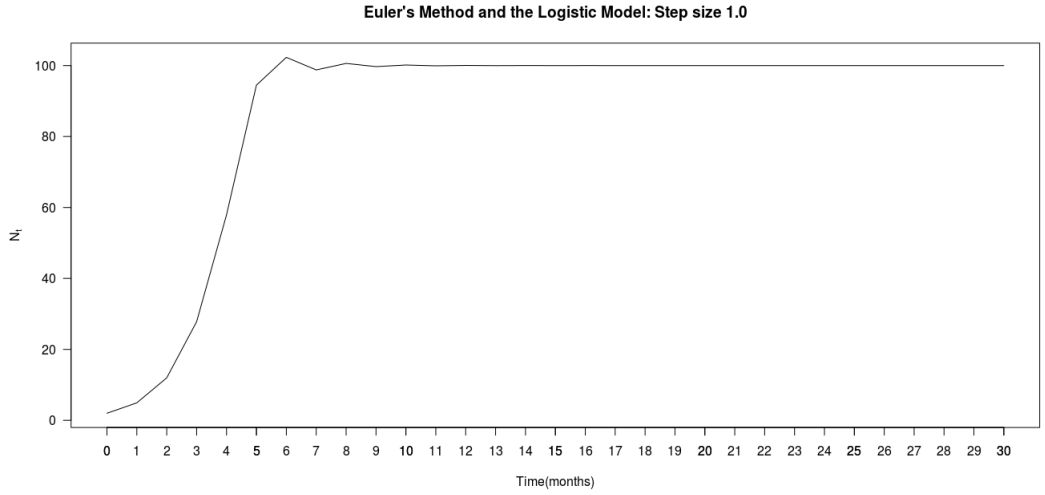
2.0.1 Step Size 0.1



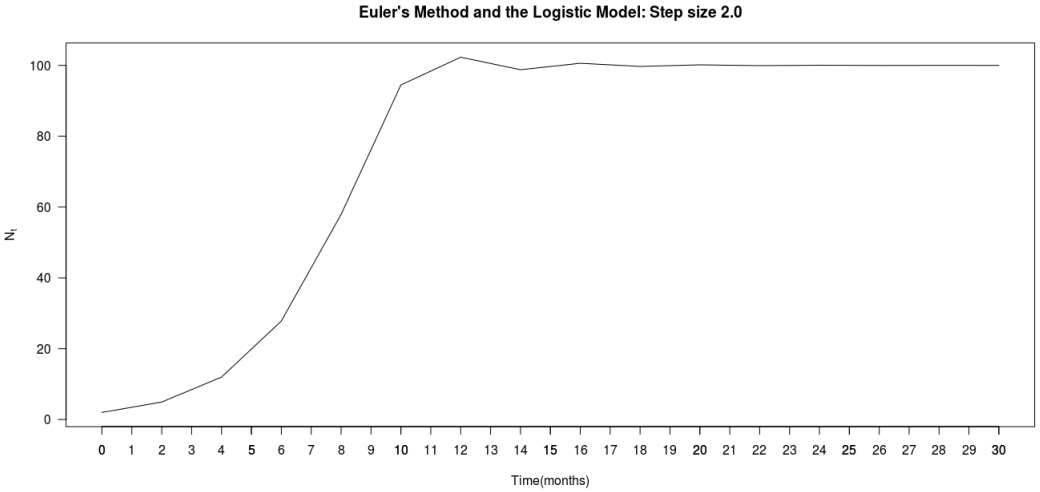
2.0.2 Step Size 0.5



2.1 Step Size 1.0



2.2 Step Size 2.0



3 Conclusion

From observing the graphs, as the stepsize increases so does the time it takes for the population to reach its carrying capacity. For example, in the graph where the stepsize is set to 0.5, it reaches its' carrying capacity in about 2.5 months. Whereelse, the graph where the stepsize is set to 1.0, its' population reaches carrying capacity in about 5 months.

4 Appendix

4.0.1 C Code for Euler's Method and the Logistic Method

```
1  /**
2   * Name: Timmy Nguyen
3   * Program: Euler's Method and the Logistic Model
4   * BIOL 480 SPRING 2017
5   * DATE: 3 - 13 - 17
6   */
7
8
9  #include <stdio.h>
10 #include <string.h>
11
12
13 int main()
14 {
15     // Initializing variables
16     float dn,dt;
17
18
19     // Initializing the rate of growth and carrying capacity
20     float r = 1.5;
21     float K = 100;
22
23
24     printf("Enter step size: \n");
25     scanf("%f", &dt);
26
27
28     /**
29      * PRECONDITION: Prompt user for time input
30      * POSTCONDITION: Read the user input
31      */
32     float time_;
33
34     printf("Enter time(in months): \n");
35     scanf("%f" ,&time_);
36
37
38     // Sets the stepsize
39     int steps = time_ / dt;
40
41
```

```
42      // Creating arrays of uknown size
43      float N[steps + 1];
44      float t[steps + 1];
45
46
47      N[0] = 2;
48      t[0] = 0;
49
50      /**
51       *
52       * Logistic Formula
53       *
54       * N_t+1 = N_t + r * N * ( 1 - ( N / K) )
55       *
56       * dn = r * N * ( 1 - ( N / K) ) * dt
57       *
58       */
59
60      for (int i = 0; i < steps; i++)
61      {
62          dn = r * N[i] * ( 1 - (N[i] / K));
63          N[i + 1] = N[i] + dn;
64          t[i + 1] = t[i] + dt;
65       }
66
67
68      // Output the results
69
70      for (int i = 0; i <= steps; i++)
71      {
72        printf("time: %.2f \t population: %.2f\n",t[i],N[i] );
73      }
74
75      /**
76       * FILE SAVE SNIPPET
77       */
78
79      char text[10];
80      char filename[100];
81      FILE *outfile;
82
83      printf( "Save data to disk file ? (y/n) : " );
84      scanf( "%s", text );
85      if ( strcmp( text, "y" ) == 0 )
86      {
```

```
87          printf( "Enter filename for dataset : " );
88          scanf( "%s", filename );
89          outfile = fopen( filename, "w" );
90          for ( int i = 0; i <= steps; i++ )
91          {
92              fprintf( outfile, "%4.1f \t%7.4f\n", t[i], N[i]);
93          }
94          fclose( outfile );
95      }
96
97  }
```

4.0.2 R Code for Generating Time vs Population

```
1
2  mydata = read.table(file.choose())
3  colnames(mydata)<-c("Time(months)", "Population")
4
5  plot(mydata[,1:2], type="l",main="Euler's Method and the
       Logistic Model: Step size 2.0"
6        ,xlab="Time(months)",ylab =expression("N"[t]), las=1)
7  axis(side=1, at=c(0:30))
```