Lab 6

Problem **1. Round-off errors**

```
> a:=evalf(25^(1/8)); a^8;
```

$$a := 1.495348781$$

$$24.99999997 \tag{1}$$

Problem **2. The logistic map: from order to chaos**

The map x->lambda*x(1-x) is called the logistic map. We would like to study the sequences defined by x(k+1)=lambda*x(k)*(1-x(k)), for any natural k, when the initial value x(0) is given.
We will take different values from x(0) in the interval (0,1).
We will take different values for lambda in the inteval [1,4].
It can be proved very easy, that, in these situations, x(k) is in the interval (0,1] for any natural k.

lambda=1

`> solve(x*(1-x)=x,x);` It has only one fixed point, 0.

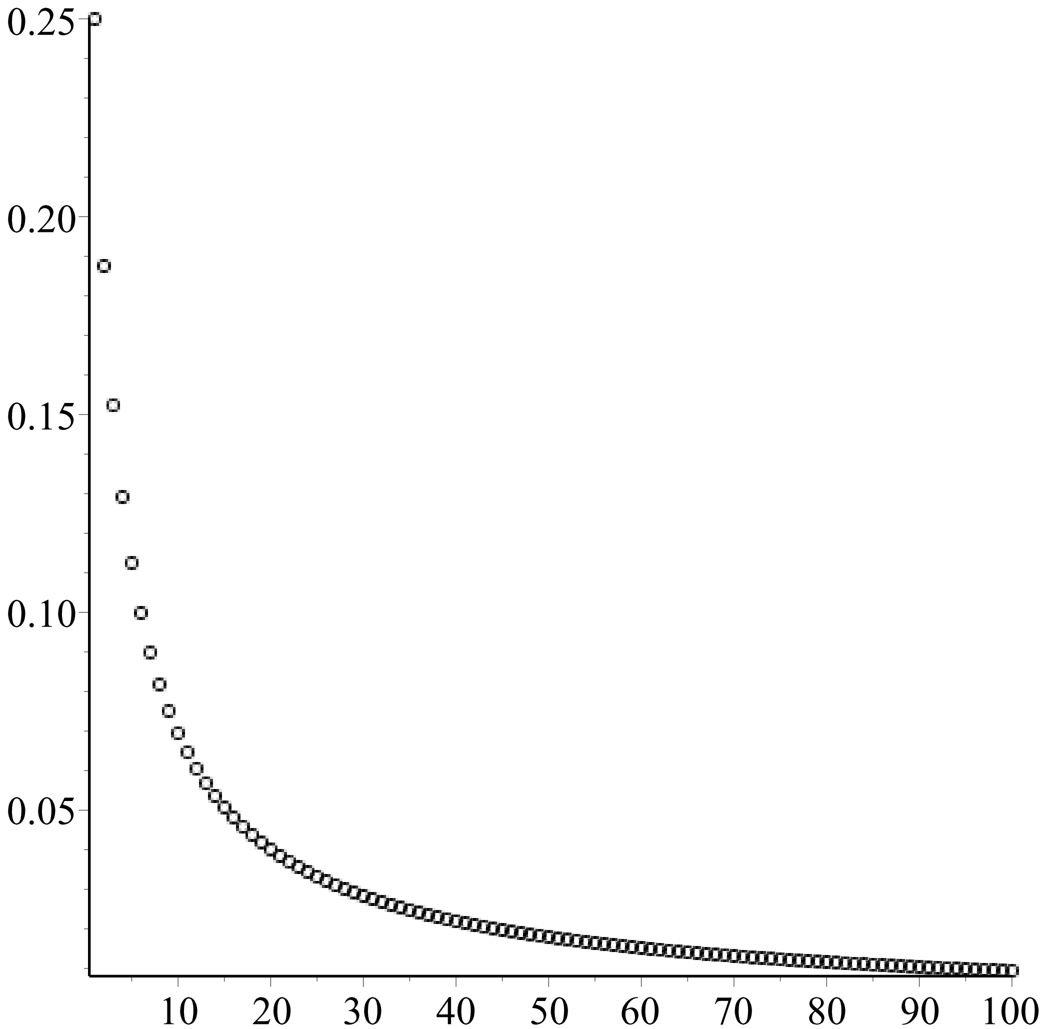
$$0, 0 \tag{2}$$

`> x:=0.5;` We set the initial value.

$$x := 0.5 \tag{3}$$

`> for i from 1 to 100 do x:=x*(1-x); f[i]:=x: od:` We compute the first 100 terms of the sequence defined by `x(k+1)=x(k)*(1-x(k))` when starting from x(0)=0.5. We do not want to see the values, but we asked Maple to save them in the vector f with 100 components.

`> points:=[[k,f[k]]$k=1..100]:with(plots):pointplot(points,symbol=circle);` The first 100 values of the sequence are seen on the vertical, represented as the rank k is increasing to the right. It seems that this sequence is monotonically decreasing to 0.

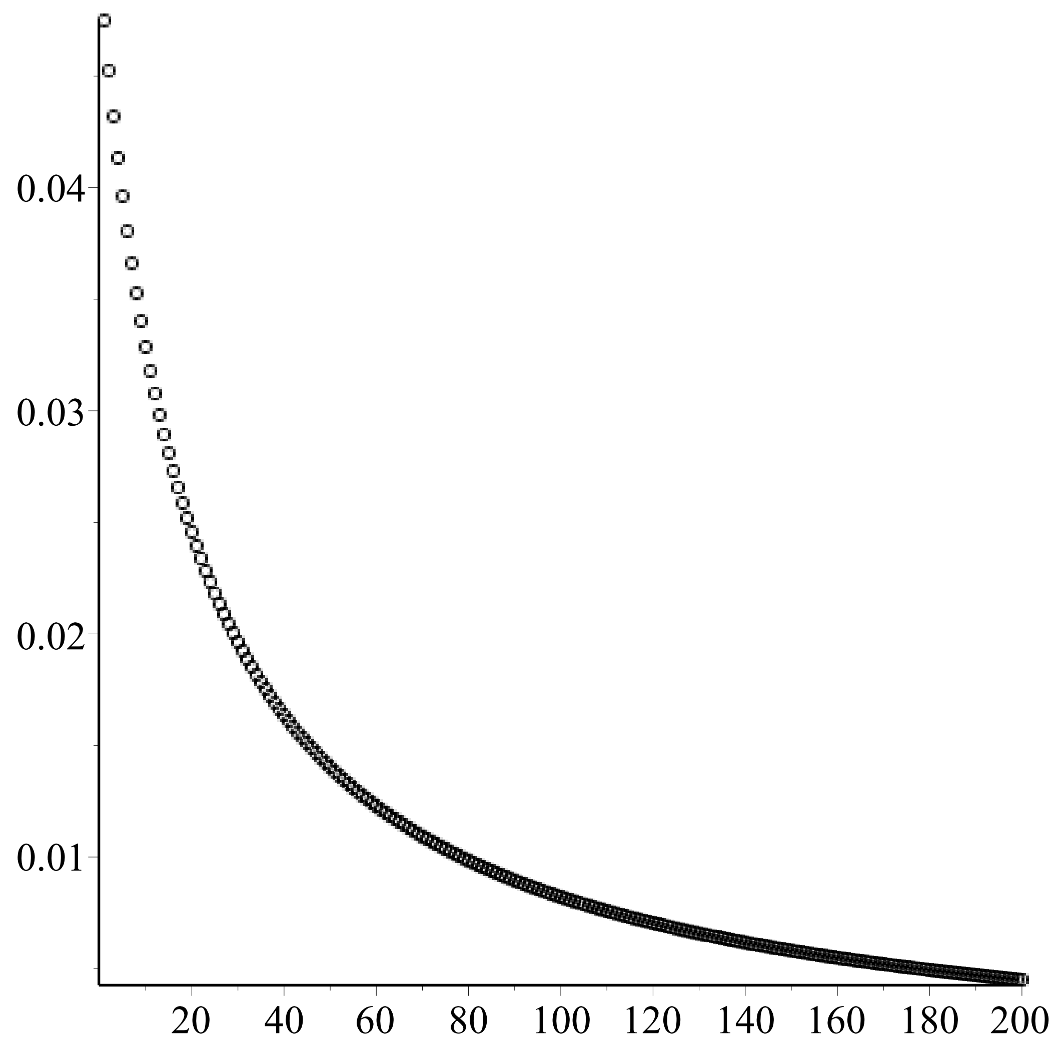



```
> x:=0.95;
```

$$x := 0.95 \quad (4)$$

```
> for i from 1 to 200 do x:=x*(1-x); f[i]:=x: od:
```

```
> points:=[[k,f[k]]$k=1..200]:with(plots):pointplot(points,symbol=
  circle);
```



> It can be proved that, when lambda=1, for any initial value x(0) in the interval (0,1), the sequence converges to 0

lambda=2

> `restart:` We want to use the same letters to denote our variables, that is why we prefer to clear the memory.

> `solve(2*x*(1-x)=x,x);` It has two fixed points, 0 and 1/2.

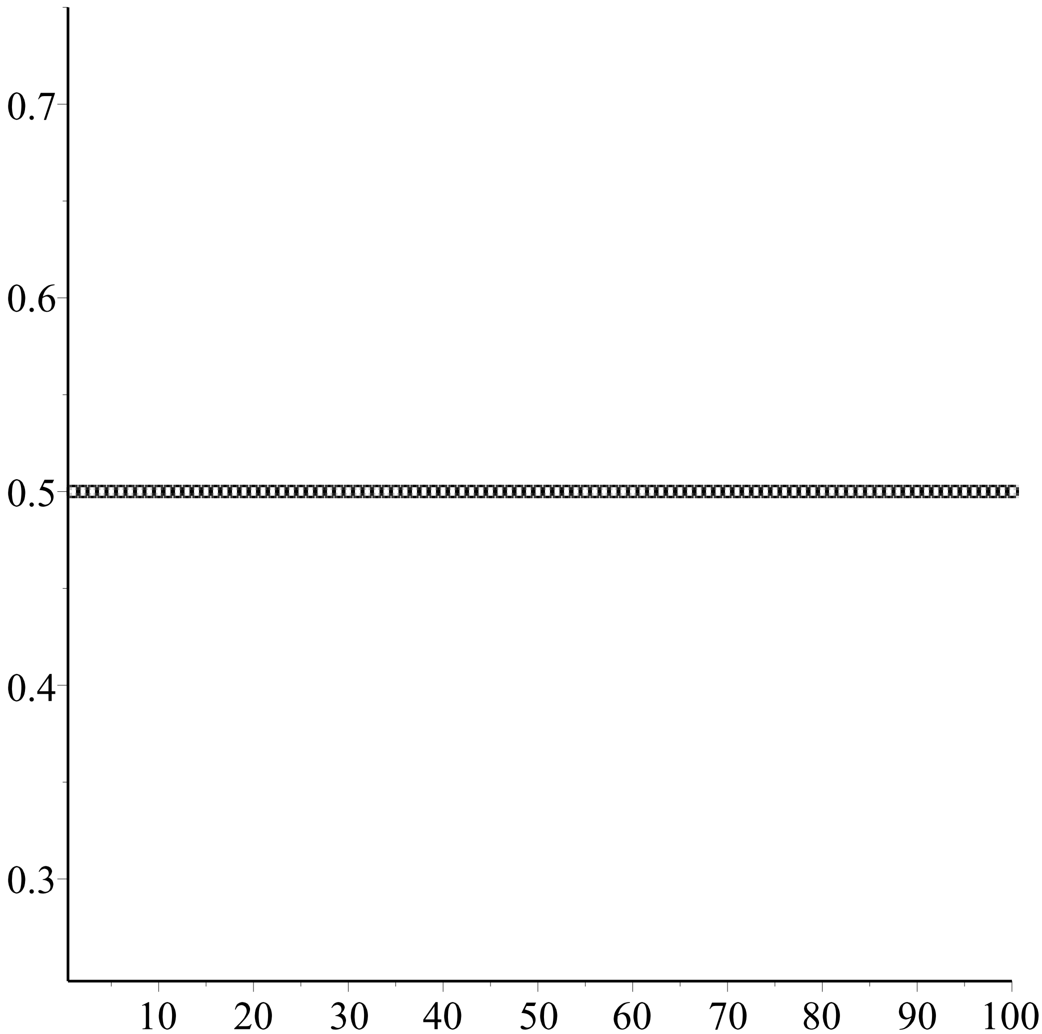
$$0, \frac{1}{2} \tag{5}$$

> `x:=0.5;` We set the initial value.

$$x := 0.5 \tag{6}$$

> `for i from 1 to 100 do x:=2*x*(1-x); f[i]:=x: od:` We compute the first 100 terms of the sequence defined by `x(k+1)=2*x(k)*(1-x(k))` when starting from x(0)= 0.5. We do not want to see the values, but we asked Maple to save them in the vector f with 100 components.

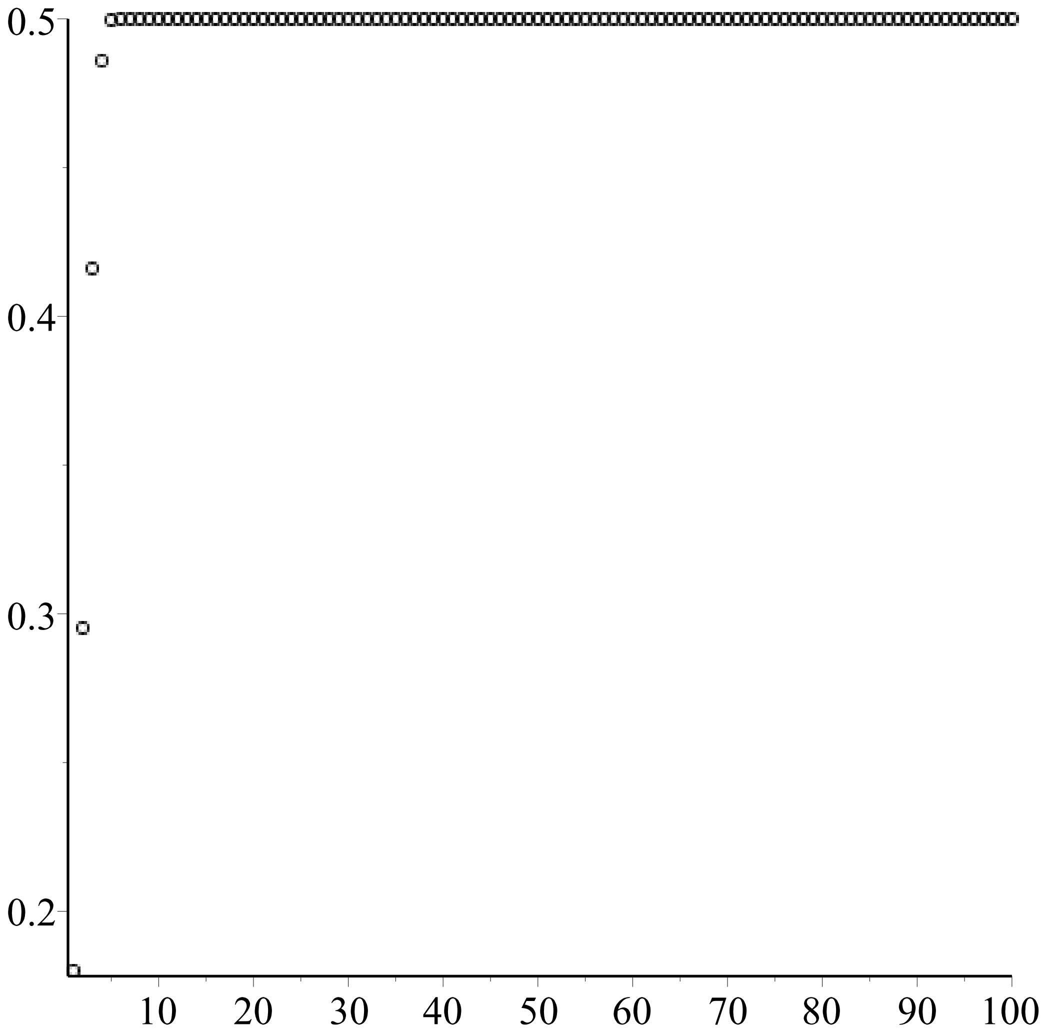
> `points:=[[k,f[k]]$k=1..100]:with(plots):pointplot(points,symbol=circle);` Now 0.5 is a fixed point, hence the sequence that starts with 0.5 is constant.




```
> x:=0.1;
```
We set another initial value.

$$x := 0.1 \quad (7)$$

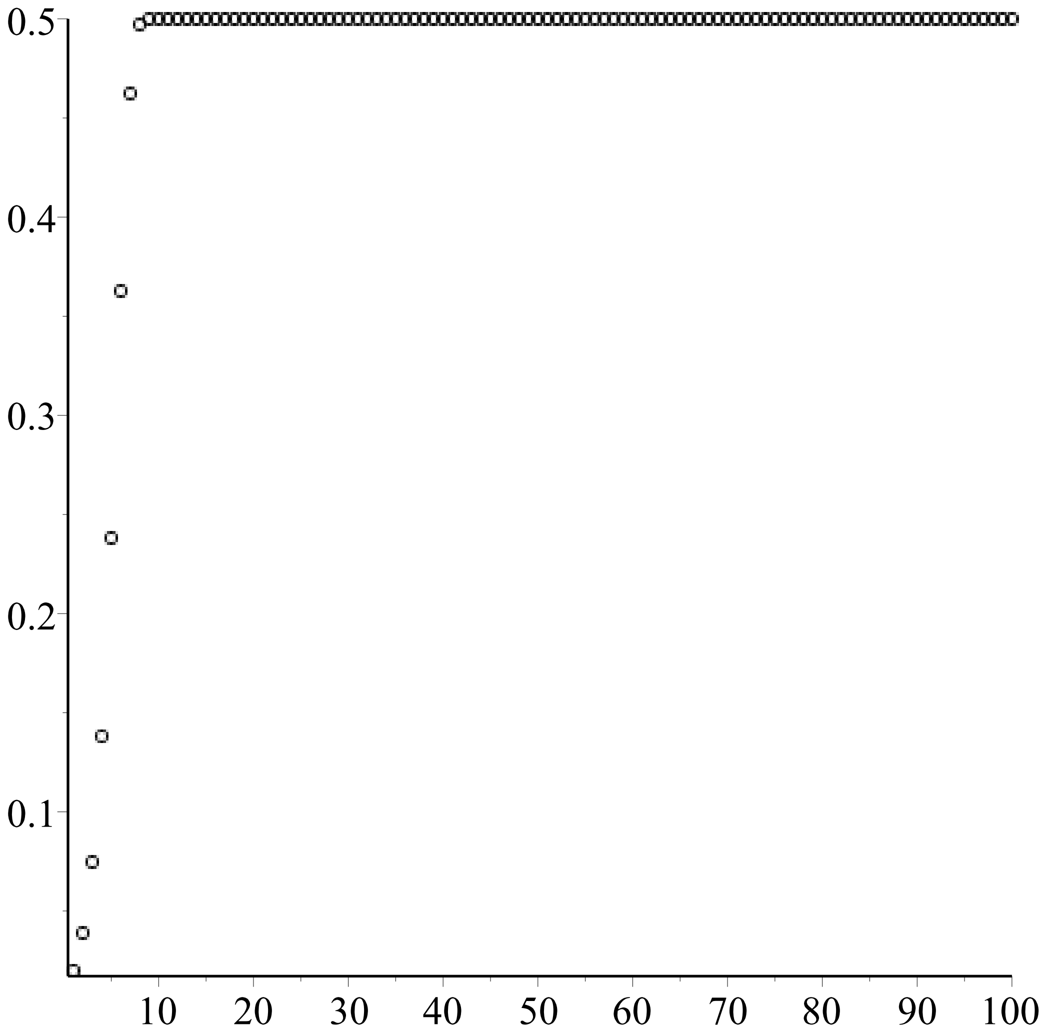
```
> for i from 1 to 100 do x:=2*x*(1-x); f[i]:=x: od:
> points:=[[k,f[k]]$k=1..100]:with(plots):pointplot(points,symbol=circle);
```
It seems that this sequence converges very very fast to 0.5. It can be proved that any element of this sequence is different from 0.5, though is very close, of course.



```
> x:=0.01;
```

$$x := 0.01 \tag{8}$$

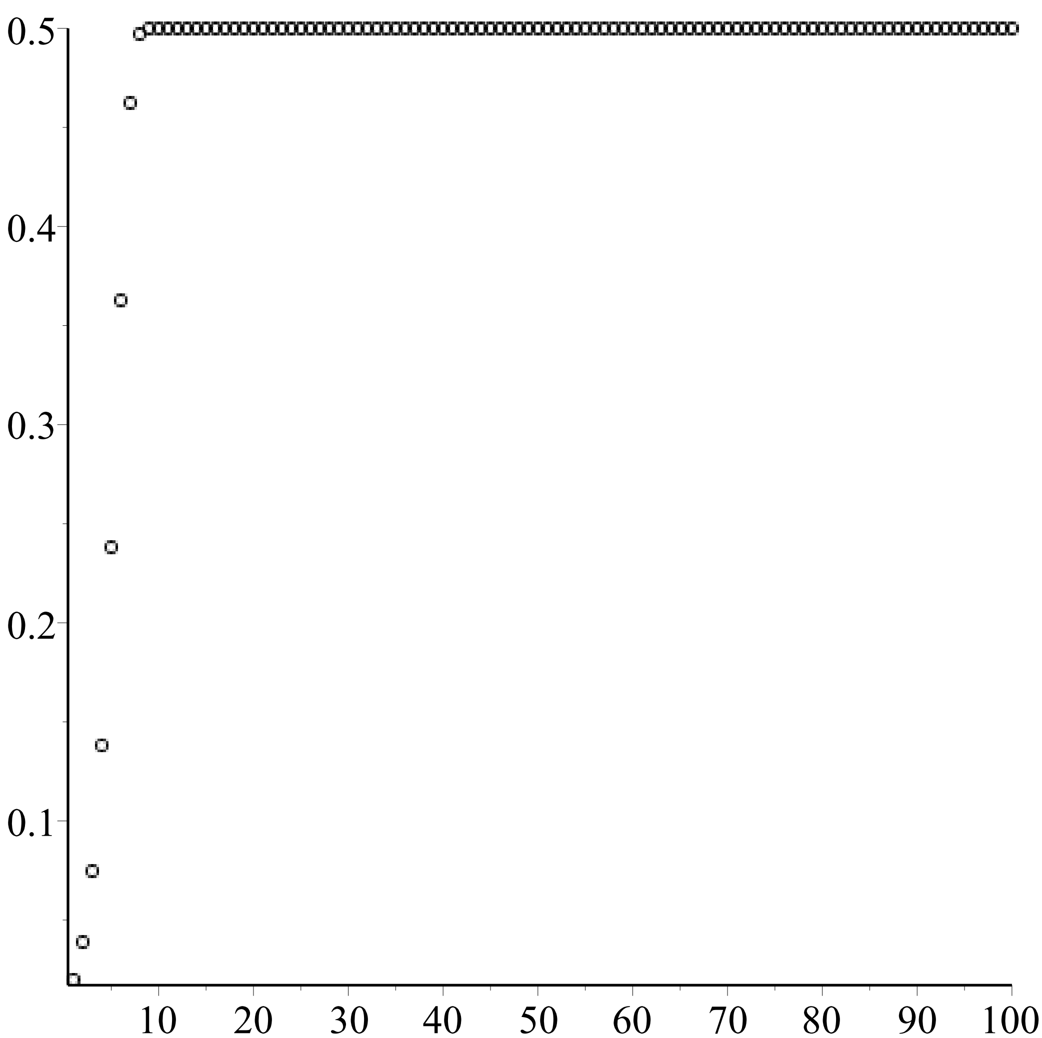
```
> for i from 1 to 100 do x:=2*x*(1-x); f[i]:=x: od:
> points:=[[k,f[k]]$k=1..100]:with(plots):pointplot(points,symbol=
  circle);
```




```
> x:=0.99;
```

$$x := 0.99 \tag{9}$$

```
> for i from 1 to 100 do x:=2*x*(1-x); f[i]:=x: od:
> points:=[[k,f[k]]$k=1..100]:with(plots):pointplot(points,symbol=
  circle);
```



> *It can be proved that, when* lambda = 2, **for** *any initial value* $x(0)$ **in** *the interval* $(0, 1)$, *the sequence converges* **to** 0.5

lambda=3.1

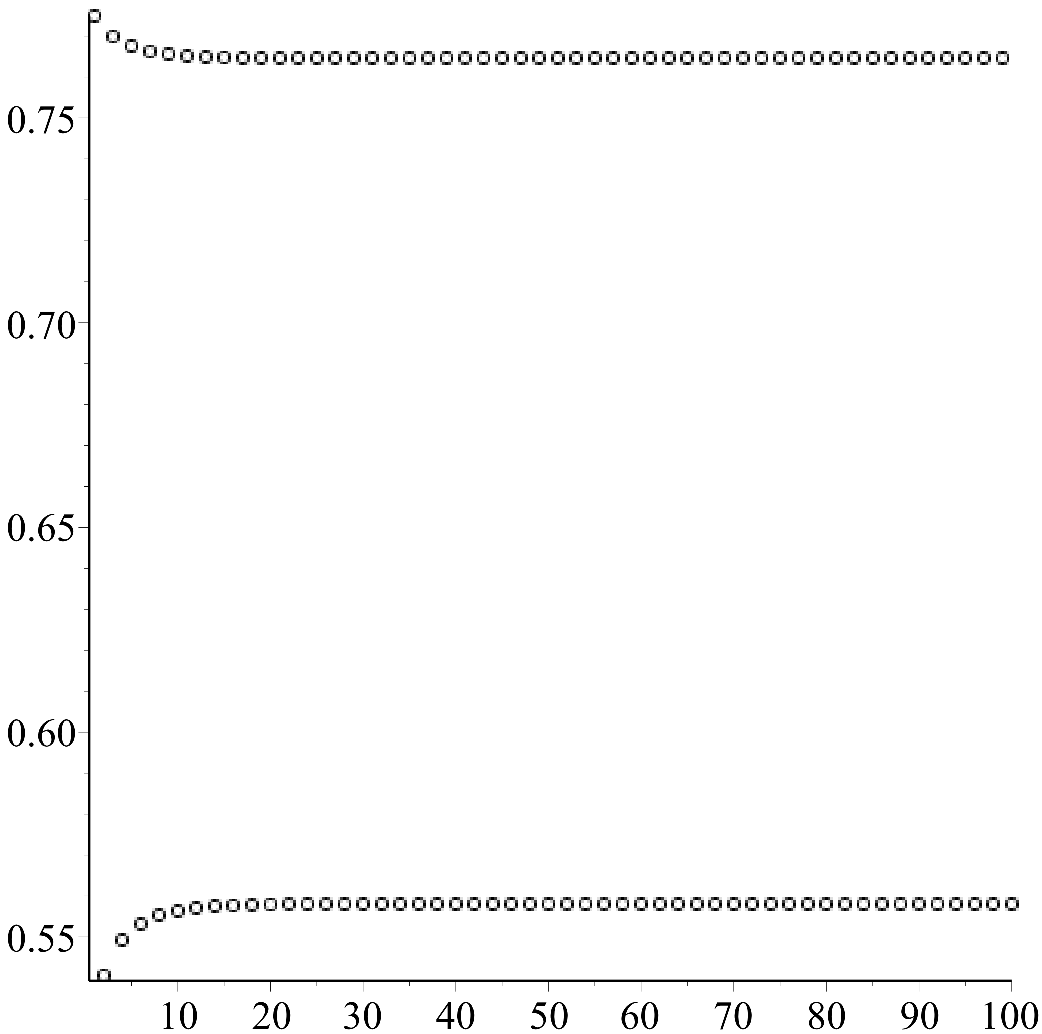
```
> restart:
> solve(3.1*x*(1-x)=x,x);
```
It has two fixed points, 0 and 0.677... .

0., 0.6774193548 **(10)**

```
> x:=0.5;
```
$x := 0.5$ **(11)**

```
> for i from 1 to 100 do x:=3.1*x*(1-x); f[i]:=x: od:
```
We compute the first 100 terms of the sequence defined by `x(k+1)=3.1*x(k)*(1-x(k))` when starting from x(0)=0.5.

```
> points:=[[k,f[k]]$k=1..100]:with(plots):pointplot(points,symbol=circle);
```
It seems that the sequence has two convergent subsequences with different limits. This indicates the presence of a 2-periodic orbit which is asymptotically stable.

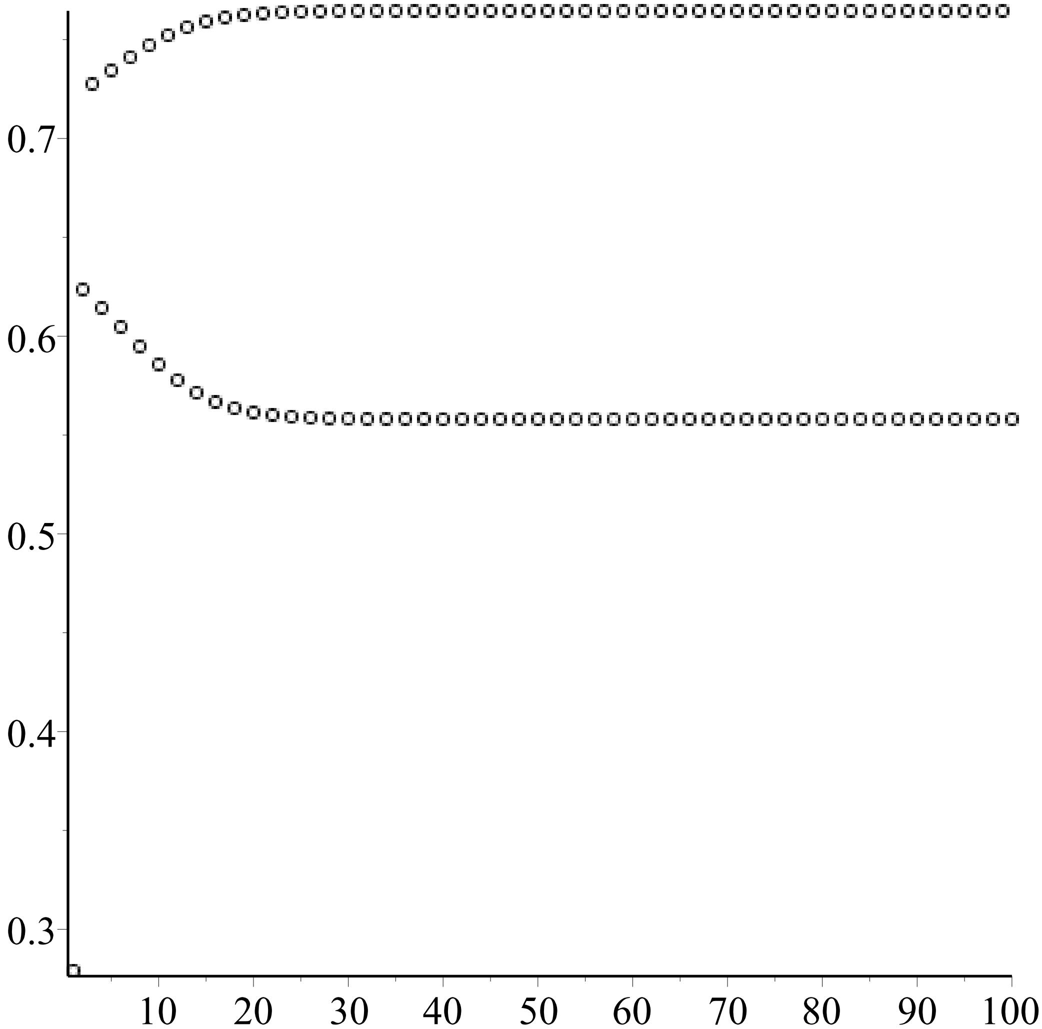



```
> x:=0.1;
```

$$x := 0.1 \tag{12}$$

```
> for i from 1 to 100 do x:=3.1*x*(1-x); f[i]:=x: od:
> points:=[[k,f[k]]$k=1..100]:with(plots):pointplot(points,symbol=
circle);
```

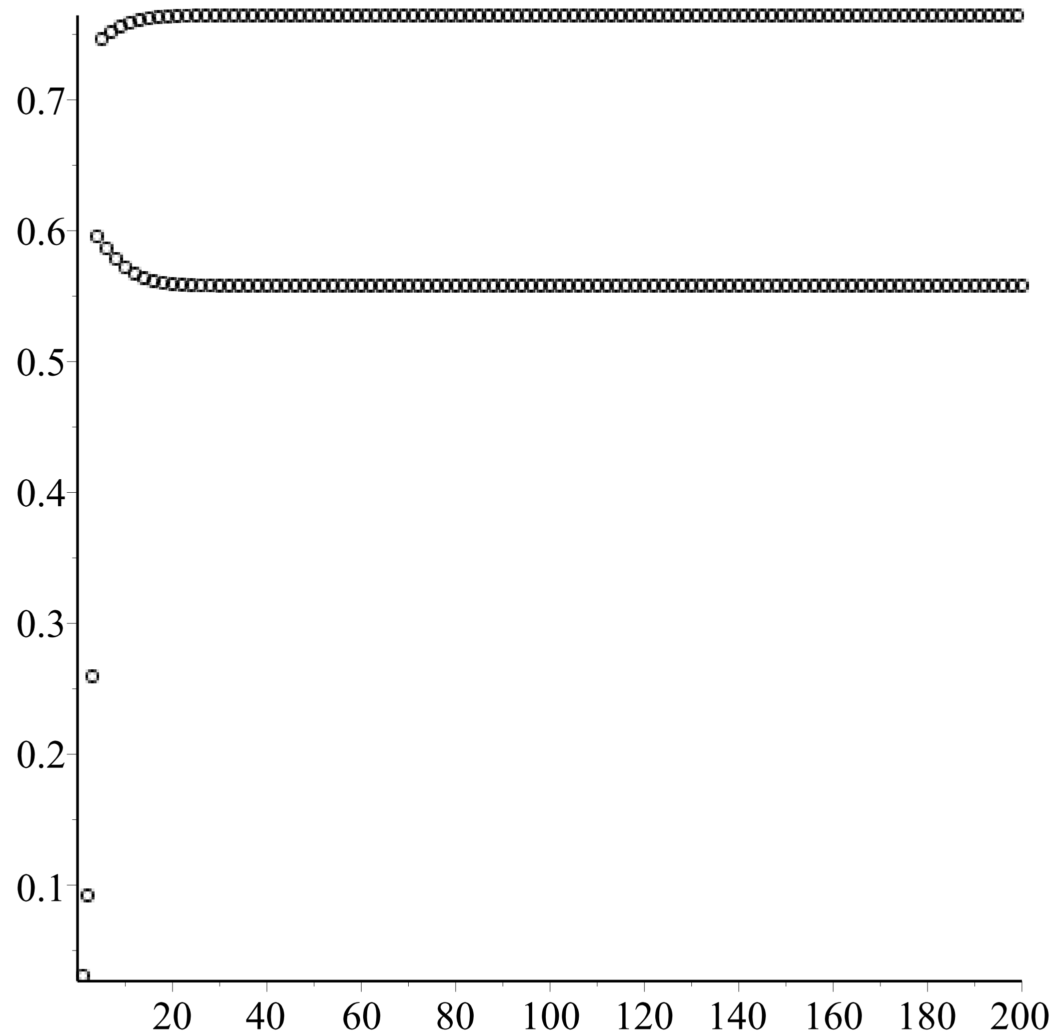
It seems that limit points of this new the sequence are the same as the limit points of the previous sequence.



```
> x:=0.99;
```

$$x := 0.99 \quad (13)$$

```
> for i from 1 to 200 do x:=3.1*x*(1-x); f[i]:=x: od:
> points:=[[k,f[k]]$k=1..200]:with(plots):pointplot(points,symbol=
  circle);
```



> It can be proved that, when lambda=3.1, for almost any initial value x(0) in the interval (0,1), the sequence has 2 convergent subsequences, whose limits do not depend on the initial value.

lambda=3.5

```
> restart:
> solve(3.5*x*(1-x)=x,x);
```

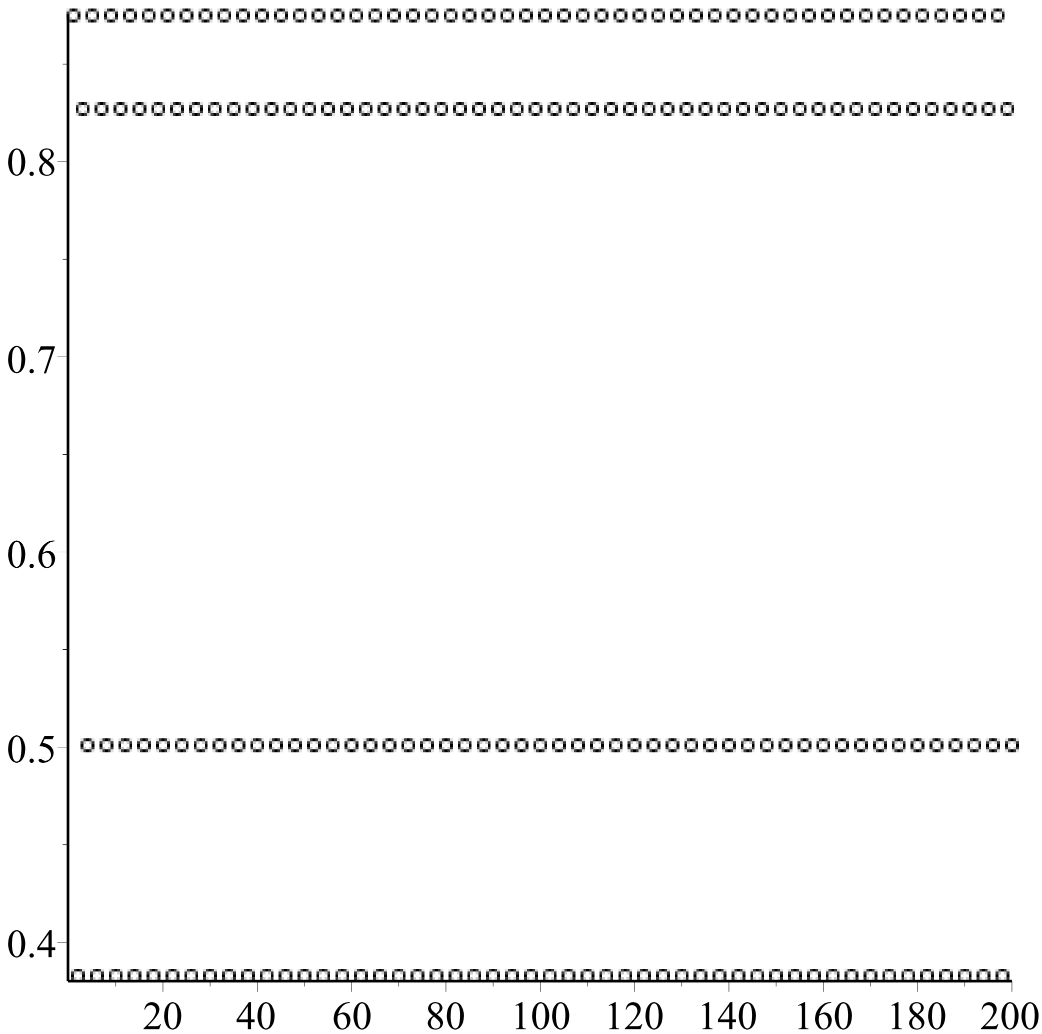
0., 0.7142857143 **(14)**

```
> x:=0.5;
```

$x := 0.5$ **(15)**

```
> for i from 1 to 200 do x:=3.5*x*(1-x); f[i]:=x: od:
> points:=[[k,f[k]]$k=1..200]:with(plots):pointplot(points,symbol=circle);
```

It seems that the sequence has 4=2^2 convergent subsequences with different limits. This indicates the presence of a 4-periodic orbit which is asymptotically stable.

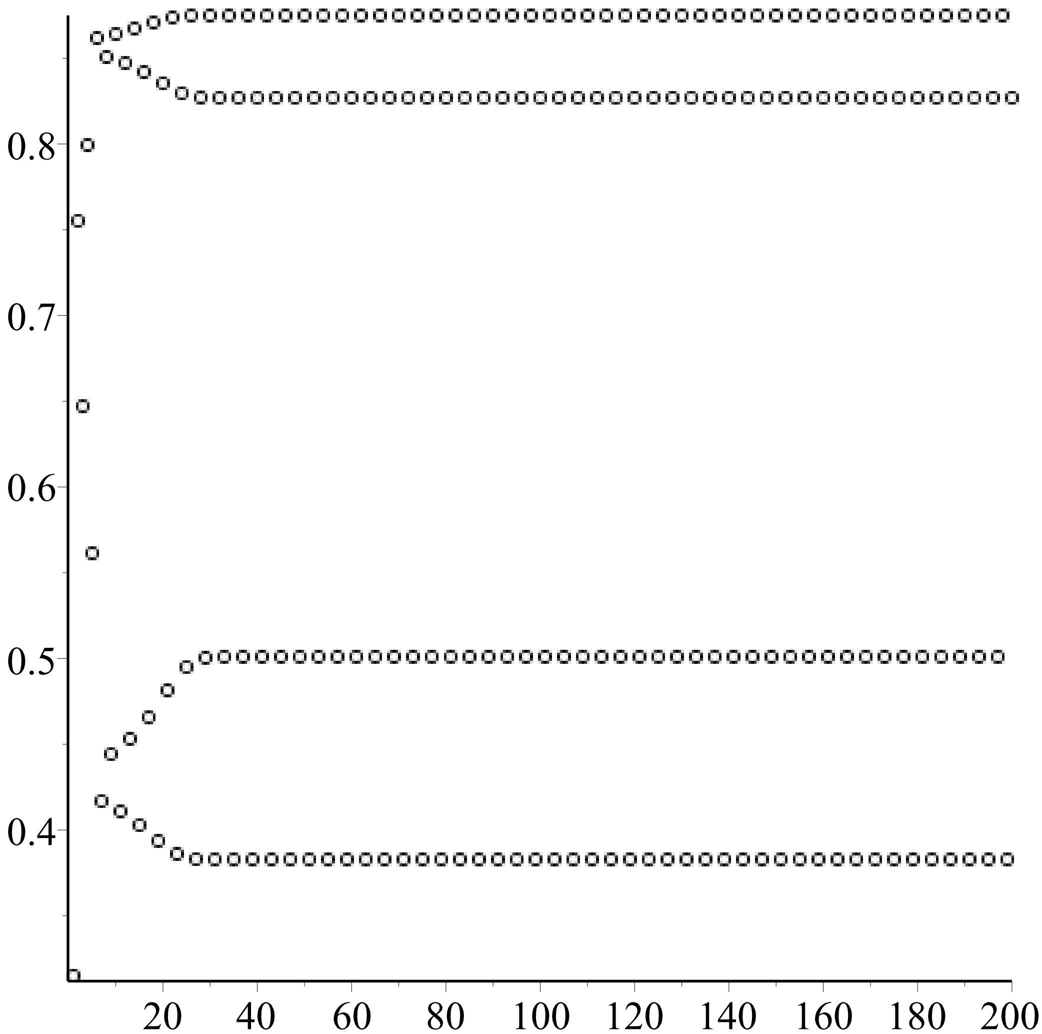



```
> x:=0.1;
```

$$x := 0.1 \qquad (16)$$

```
> for i from 1 to 200 do x:=3.5*x*(1-x); f[i]:=x: od:
> points:=[[k,f[k]]$k=1..200]:with(plots):pointplot(points,symbol=
circle);
```

It seems that limit points of this new the sequence are the same as the limit points of the previous sequence.



It can be proved that, when lambda=3.5, for almost any initial value x(0) in the interval (0,1), the sequence has 4 convergent subsequences, whose limits do not depend on the initial value.

lambda=3.55

```
> restart:
> solve(3.55*x*(1-x)=x,x);
```

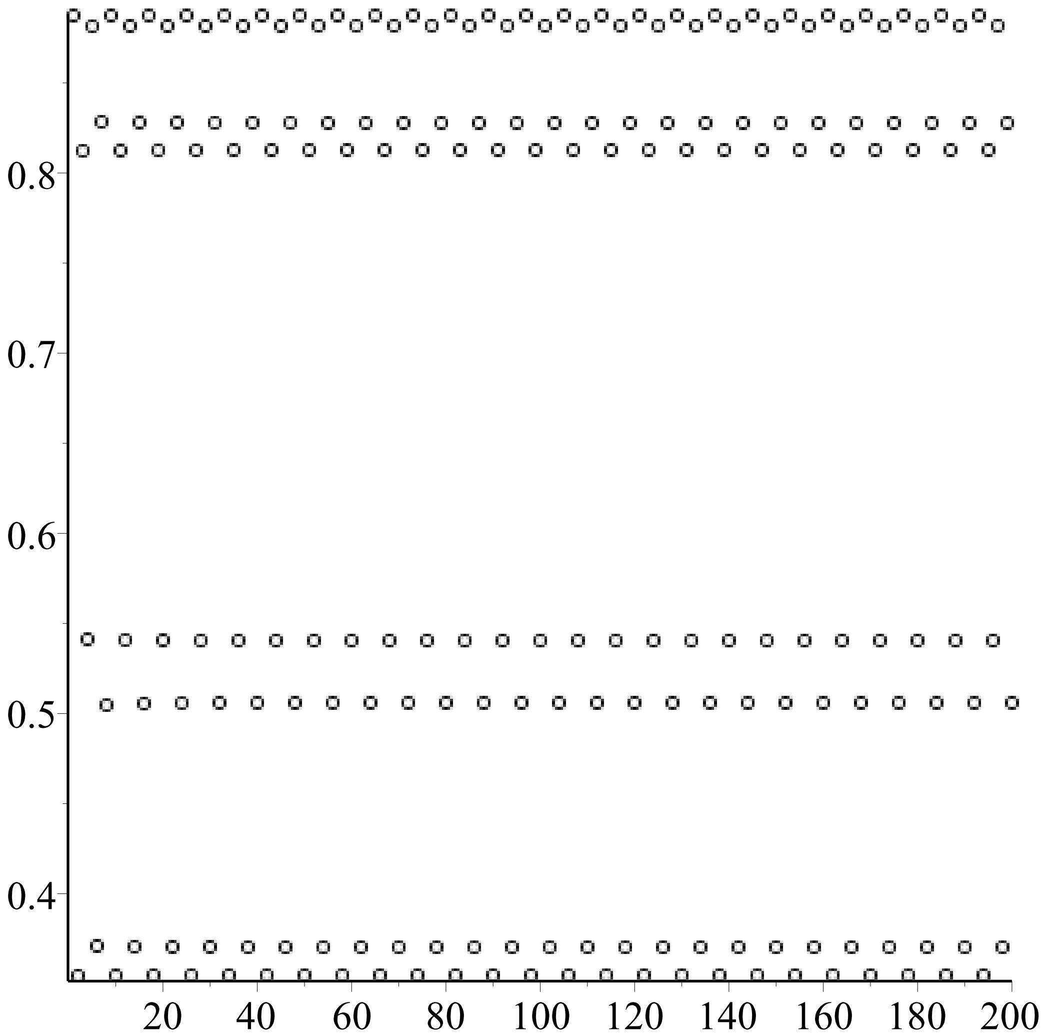
0., 0.7183098592 **(17)**

```
> x:=0.5;
```

$x := 0.5$ **(18)**

```
> for i from 1 to 200 do x:=3.55*x*(1-x); f[i]:=x: od:
> points:=[[k,f[k]]$k=1..200]:with(plots):pointplot(points,symbol=
circle);
```

It seems that the sequence has 8=2^3 convergent subsequences with different limits. This indicates the presence of a 8-periodic orbit which is asymptotically stable.

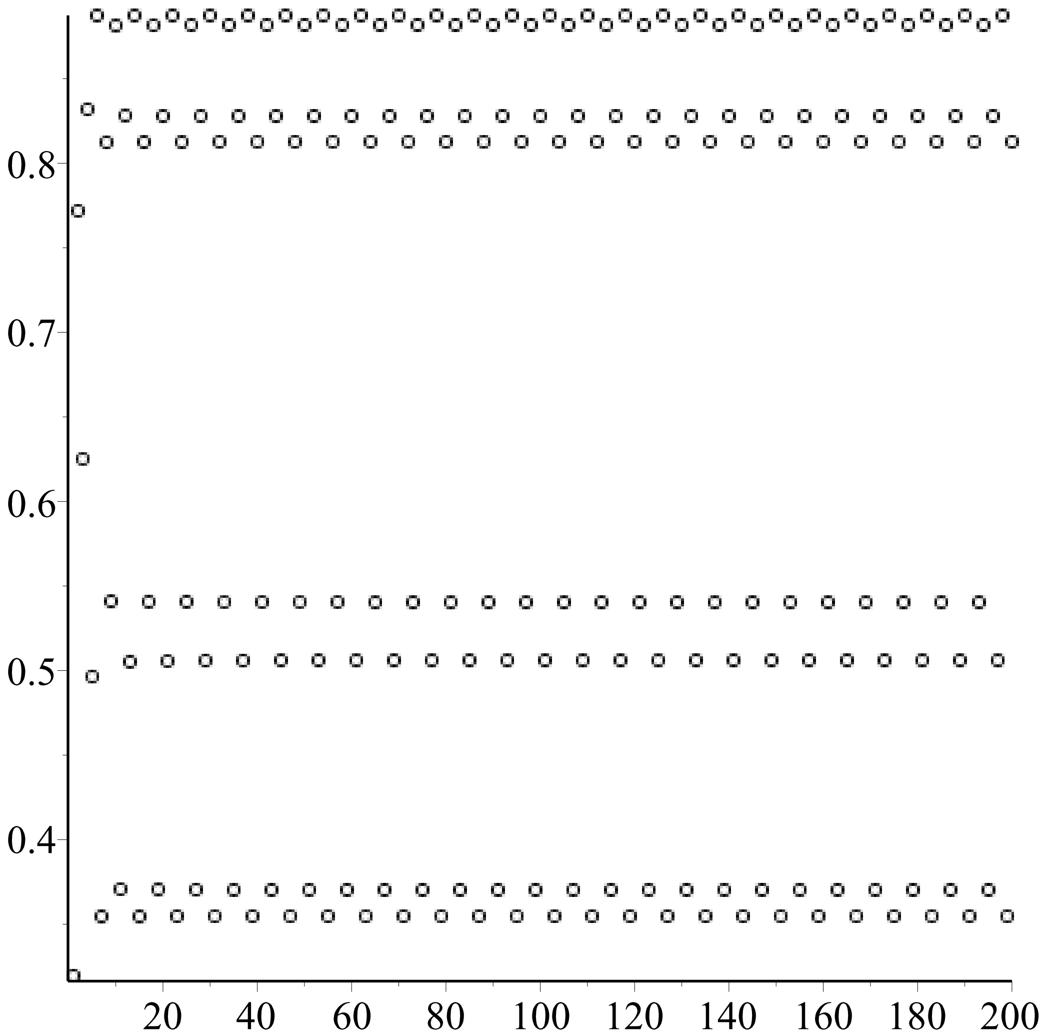



```
> x:=0.1;
```

$$x := 0.1 \quad \textbf{(19)}$$

```
> for i from 1 to 200 do x:=3.55*x*(1-x); f[i]:=x: od:
> points:=[[k,f[k]]$k=1..200]:with(plots):pointplot(points,symbol=
circle);
```

It seems that limit points of this new the sequence are the same as the limit points of the previous sequence.



It can be proved that, when lambda=3.55, for almost any initial value x(0) in the interval (0,1), the sequence has 8 convergent subsequences, whose limits do not depend on the initial value.

lambda=3.6

```
> restart:
> solve(3.6*x*(1-x)=x,x);
```

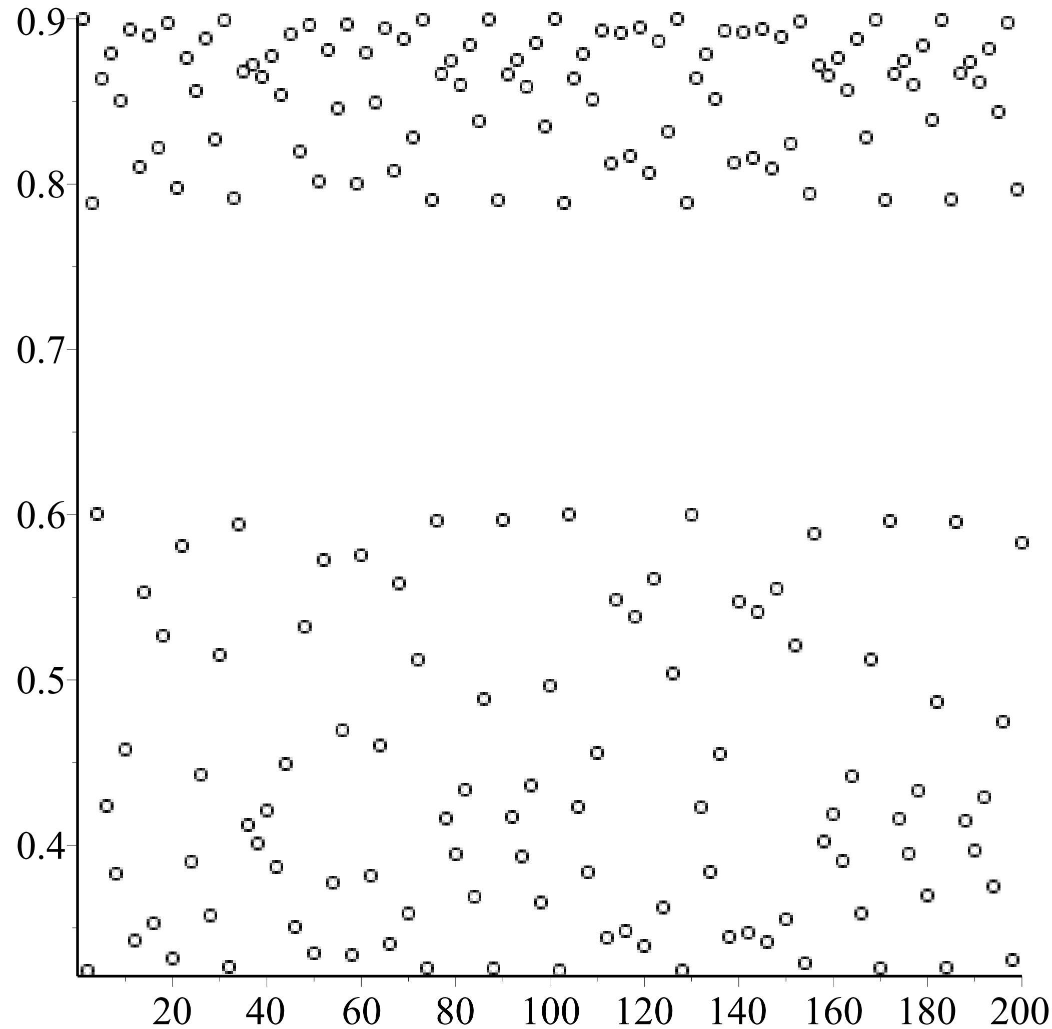
0., 0.7222222222 **(20)**

```
> x:=0.5;
```

$x := 0.5$ **(21)**

```
> for i from 1 to 200 do x:=3.6*x*(1-x); f[i]:=x: od:
> points:=[[k,f[k]]$k=1..200]:with(plots):pointplot(points,symbol=
circle);
```

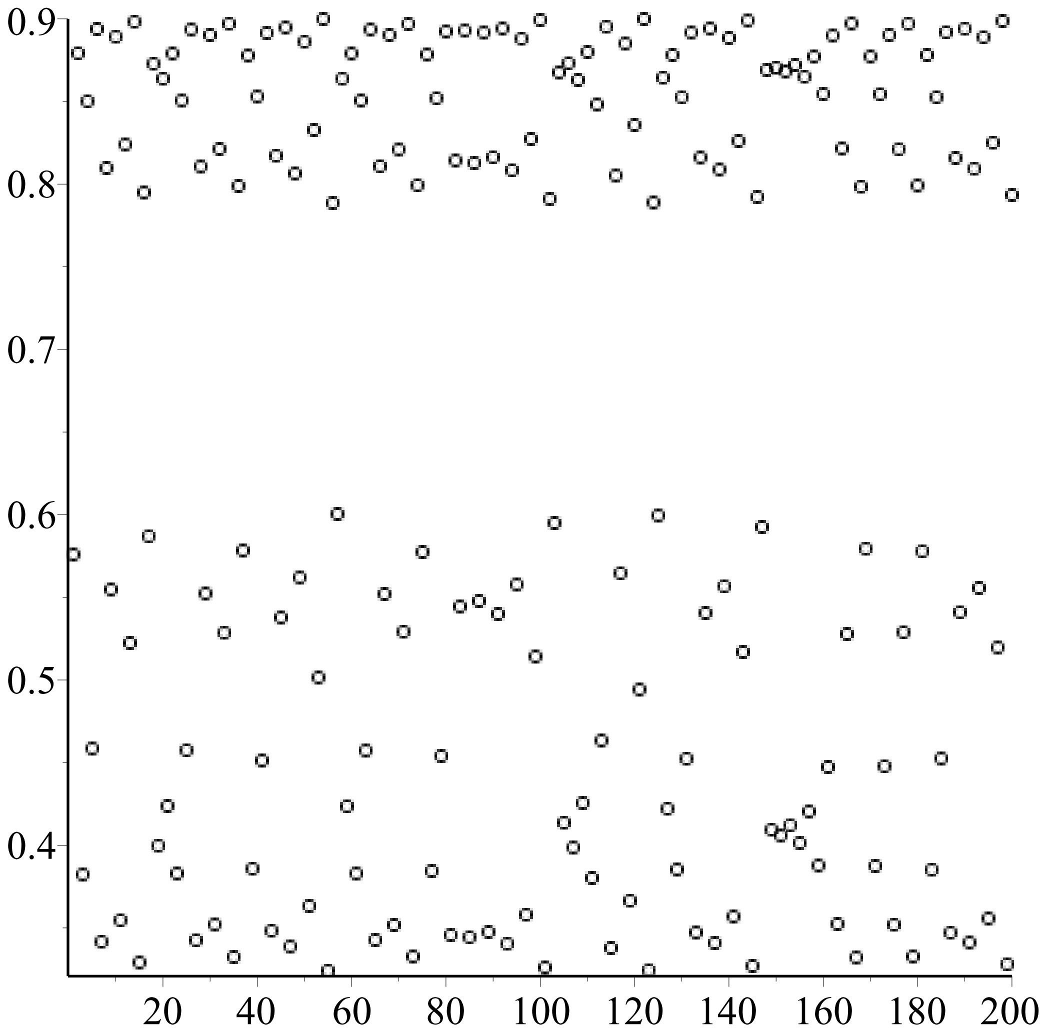
It is very difficult to see any pattern here. But at least we can say that there is no element of this sequence which is close to 0.72222.




```
> x:=0.2;
```

$$x := 0.2 \tag{22}$$

```
> for i from 1 to 200 do x:=3.6*x*(1-x); f[i]:=x: od:
> points:=[[k,f[k]]$k=1..200]:with(plots):pointplot(points,symbol=
  circle);
```




```
>
```

lambda=3.8

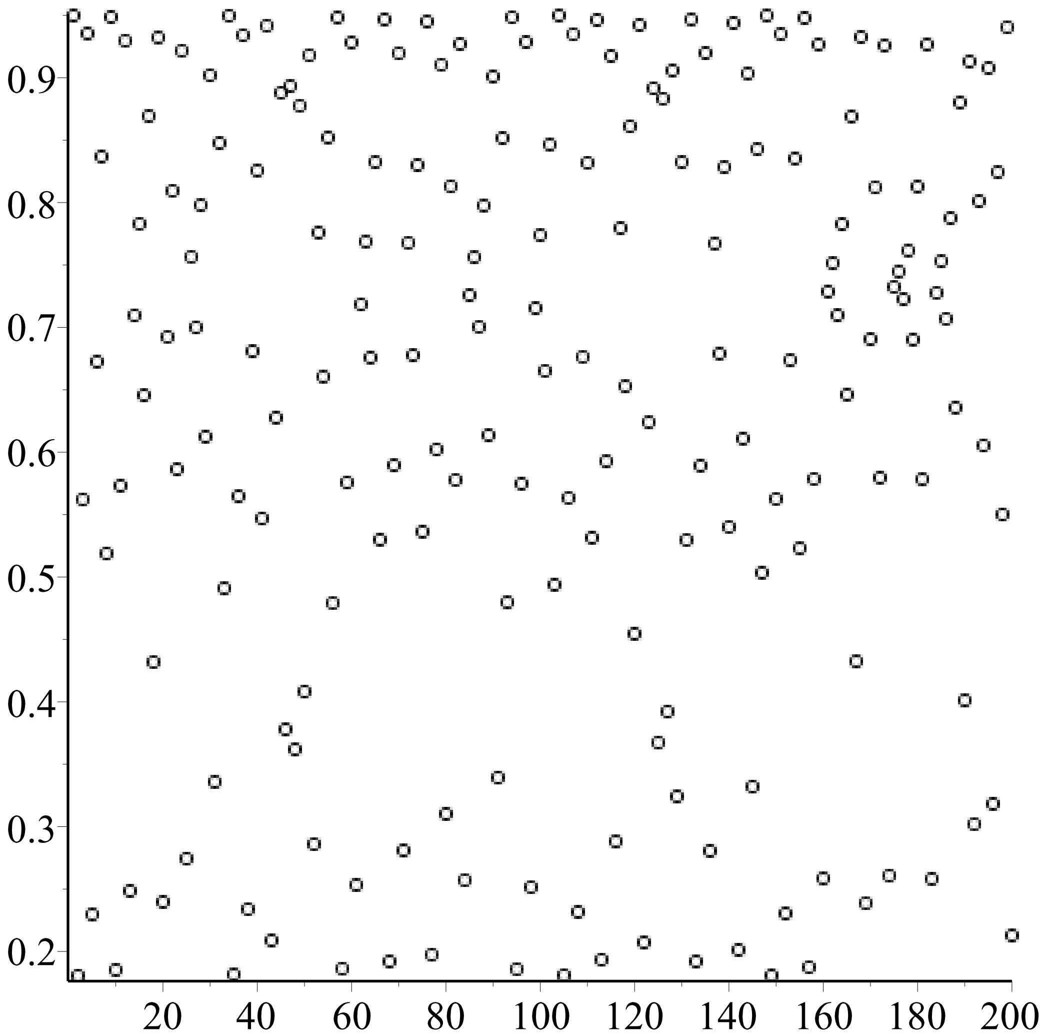
```
> restart:
> solve(3.8*x*(1-x)=x,x);
```

0., 0.7368421053 **(23)**

```
> x:=0.5;
```

$x := 0.5$ **(24)**

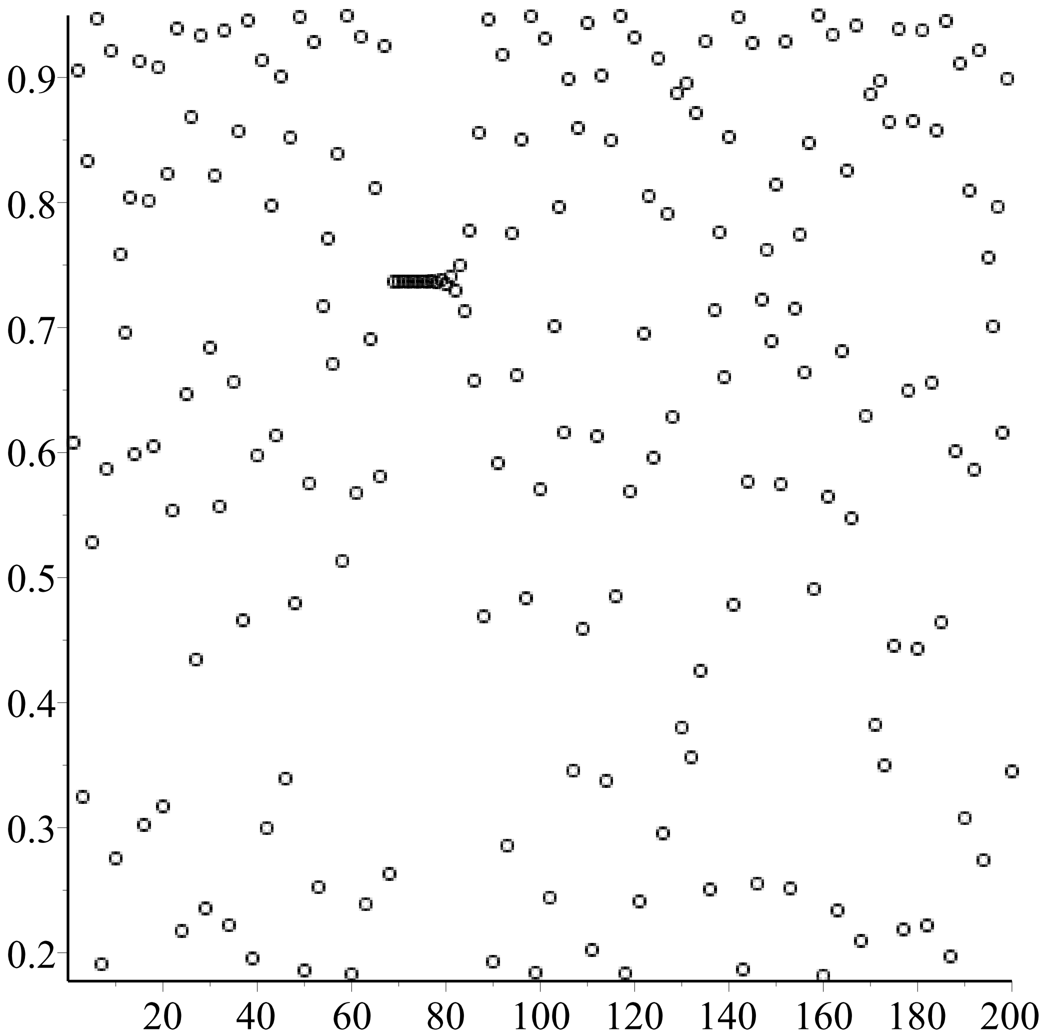
```
> for i from 1 to 200 do x:=3.8*x*(1-x); f[i]:=x: od:
> points:=[[k,f[k]]$k=1..200]:with(plots):pointplot(points,symbol=
  circle);
```




```
> x:=0.2;
```

$$x := 0.2 \tag{25}$$

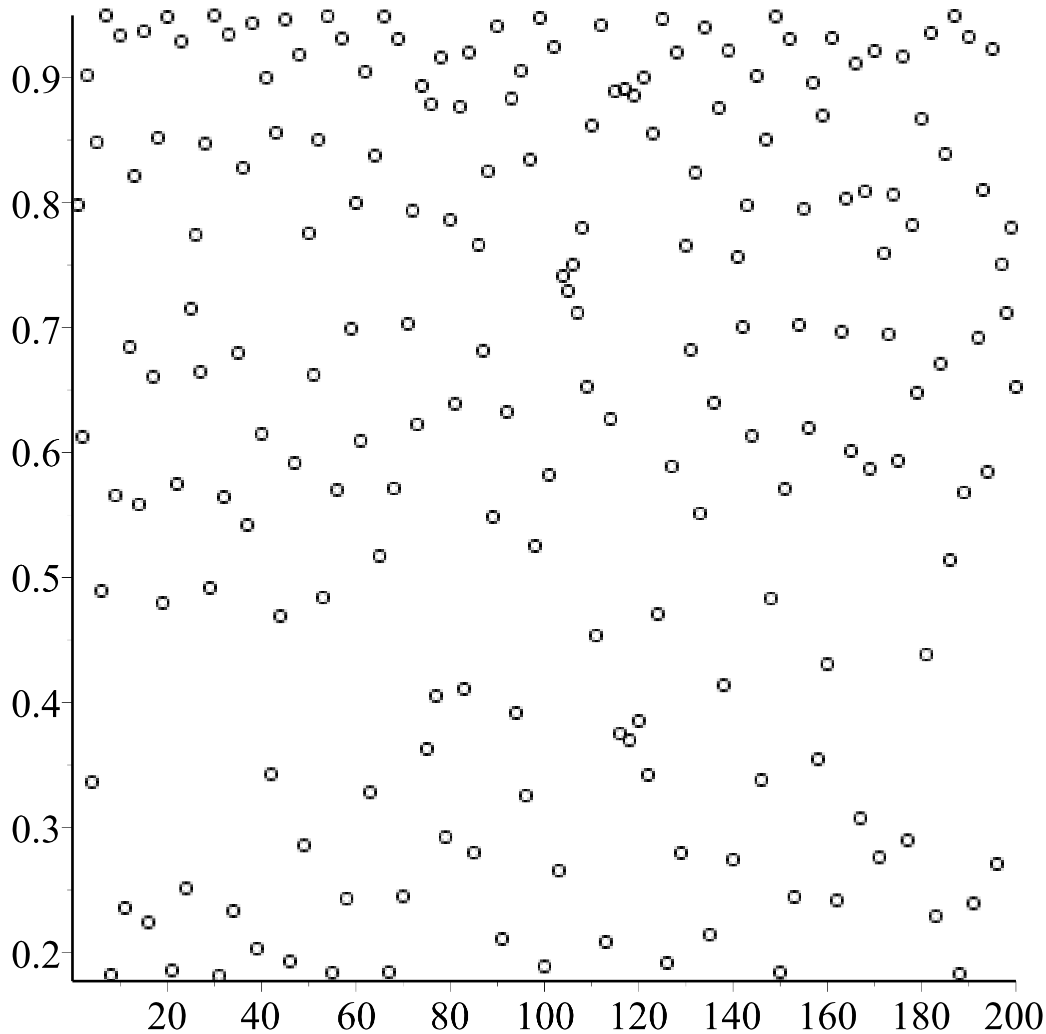
```
> for i from 1 to 200 do x:=3.8*x*(1-x); f[i]:=x: od:
> points:=[[k,f[k]]$k=1..200]:with(plots):pointplot(points,symbol=
  circle);
```




```
> x:=0.78;
```

$$x := 0.78 \tag{26}$$

```
> for i from 1 to 200 do x:=3.8*x*(1-x); f[i]:=x: od:
> points:=[[k,f[k]]$k=1..200]:with(plots):pointplot(points,symbol=
  circle);
```



> *This is chaos ...*

Problem 3: **lambda=4 or Can we trust Maple?**

```
> restart:
> x:=0.67; y:=0.67; z:=0.67;
```
We take the same initial value for three sequences defined by the same formula (so, the same sequence, in fact!)

$$x := 0.67$$
$$y := 0.67$$
$$z := 0.67 \quad \textbf{(27)}$$

```
> for i from 1 to 40 do x:=4*x*(1-x): y:=4*y-4*y^2: z:=4*z-4*z*z:
  print(i,x,y,z); od:
```

1, 0.8844, 0.8844, 0.8844

2, 0.40894656, 0.40894656, 0.40894656

3, 0.9668370844, 0.9668370843, 0.9668370843

4, 0.1282525465, 0.1282525471, 0.128252547

5, 0.4472153232, 0.4472153251, 0.4472153248

6, 0.9888551116, 0.9888551126, 0.9888551121

7, 0.04408271944, 0.04408271593, 0.044082717
8, 0.1685577332, 0.1685577203, 0.1685577242
9, 0.5605840952, 0.5605840609, 0.5605840713
10, 0.9853182696, 0.9853182860, 0.985318281
11, 0.05786470876, 0.05786464510, 0.057864665
12, 0.2180655370, 0.2180653118, 0.2180653822
13, 0.6820518344, 0.6820513264, 0.6820514851
14, 0.8674285184, 0.8674292580, 0.867429027
15, 0.4599851356, 0.4599829615, 0.459983640
16, 0.9935952424, 0.9935945465, 0.9935947637
17, 0.02545494672, 0.02545769466, 0.025456837
18, 0.09922796964, 0.09923840176, 0.09923514580
19, 0.3575271188, 0.3575605655, 0.3575501266
20, 0.9188059124, 0.9188440300, 0.9188321340
21, 0.2984064310, 0.2982787141, 0.298318574
22, 0.8374401316, 0.8372340914, 0.8372984096
23, 0.5445366304, 0.5450926701, 0.544919131
24, 0.9920659544, 0.9918666046, 0.991929087
25, 0.03148438608, 0.03226897351, 0.032023093
26, 0.1219724780, 0.1249107474, 0.1239904581
27, 0.4283807704, 0.4372322103, 0.4344672976
28, 0.9794827440, 0.9842408184, 0.9828218594
29, 0.08038519284, 0.06204331879, 0.067532208
30, 0.2956936545, 0.2327757815, 0.2518864355
31, 0.8330356688, 0.7143648682, 0.7537586364
32, 0.5563489732, 0.8161908130, 0.742426218
33, 0.9872991728, 0.6000934791, 0.764918115
34, 0.05015806476, 0.9599251820, 0.719273569
35, 0.1905689332, 0.1538753078, 0.807676408
36, 0.6170096596, 0.5207907898, 0.621340912
37, 0.9452349584, 0.9982709723, 0.941105532
38, 0.2070633273, 0.006904152853, 0.221703639
39, 0.6567524232, 0.02742594211, 0.6902045418
40, 0.9017147112, 0.1066950392, 0.855288929 **(28)**

> This is because the round − off errors are different when using a different operations order **and**, more important, they are magnified due **to** the fact that the map $x \to 4\,x\,(1-x)$ *is chaotic. Moreover, last year I obtained other values. Try yourself* **and** *compare what you obtained with the values above. The logisitic map is very well studied. There are books dedicated* **to** *this map. Ask Google about the logistic map* **and** *you can learn more interesting facts.*

>

Problem **4.** For the IVP y'=2xy, x\in[0,1], y(0)=1, first find its exact solution and then use

(a) Euler's method with step size h=0.1;   (b) the improved Euler's method with step size h=0.1;

to find approximate values of the solution in the interval [0,1]. Compute and write in your notebooks the absolute value of the
difference between the correct value and the approximate one at x=0.5 and, respectively, x=1.
Formulate a conclusion.

```
> restart:f:=(x,y)->2*x*y;sol4:=dsolve({diff(y(x),x)=f(x,y(x)),y(0)=1},{y(x)});phi:=unapply(rhs(sol4),x);
```

$$f := (x, y) \to 2\, x\, y$$

$$sol4 := y(x) = e^{x^2}$$

$$\phi := x \to e^{x^2} \quad (29)$$

```
> h:=0.1;x:=0;y:=1;
```

$$h := 0.1$$

$$x := 0$$

$$y := 1 \quad (30)$$

```
> for i from 1 to 10 do y:=evalf(y+h*f(x,y)): x:=x+h: print(x,y,phi(x),abs(y-phi(x))); od:
```

0.1, 1., 1.010050167, 0.010050167
0.2, 1.02, 1.040810774, 0.020810774
0.3, 1.0608, 1.094174284, 0.033374284
0.4, 1.124448, 1.173510871, 0.049062871
0.5, 1.21440384, 1.284025417, 0.069621577
0.6, 1.335844224, 1.433329415, 0.097485191
0.7, 1.496145531, 1.632316220, 0.136170689
0.8, 1.705605905, 1.896480879, 0.190874974
0.9, 1.978502850, 2.247907987, 0.269405137
1.0, 2.334633363, 2.718281828, 0.383648465 (31)

```
>
```

from above, the absolute value of the difference between the correct value and the approximate one at x=0.5 is 0.069621577,
while the absolute value of the difference between the correct value and the approximate one at x=1 is 0.383648465.

```
> restart:f:=(x,y)->2*x*y;sol4:=dsolve({diff(y(x),x)=f(x,y(x)),y(0)=1},{y(x)});phi:=unapply(rhs(sol4),x);
```

$$f := (x, y) \to 2\, x\, y$$

$$sol4 := y(x) = e^{x^2}$$

$$\phi := x \to e^{x^2} \quad (32)$$

```
> h:=0.1;x:=0;y:=1;
```

$$h := 0.1$$

$$x := 0$$

$$y := 1 \quad (33)$$

```
> for i from 1 to 10 do y:=y+h/2*f(x,y)+h/2*f(x+h,y+h*f(x,y)): x:=x+h: print(x,y,phi(x),abs(y-phi(x))); od:
```

0.1, 1.010000000, 1.010050167, 0.000050167
0.2, 1.040704000, 1.040810774, 0.000106774
0.3, 1.093988045, 1.094174284, 0.000186239
0.4, 1.173192779, 1.173510871, 0.000318092
0.5, 1.283472900, 1.284025417, 0.000552517
0.6, 1.432355756, 1.433329415, 0.000973659
0.7, 1.630593792, 1.632316220, 0.001722428
0.8, 1.893445511, 1.896480879, 0.003035368
0.9, 2.242596863, 2.247907987, 0.005311124
1.0, 2.709057011, 2.718281828, 0.009224817 **(34)**

```
>
```

from above, the absolute value of the difference between the correct value and the approximate one at x=0.5 is 0.000552517,
while the absolute value of the difference between the correct value and the approximate one at x=1 is 0.009224817.

Conclusion: it seems that the improved Euler's method is better than the Euler's method. It also seems that the errors increase as x increases.

Problem **5.** For the IVP y'=x^2+y^2, y(0)=0, apply both methods in the interval [0,2] with step size h= 0.1.
Use DEplot to represent the direction field of the differential equation and the graph of the solution of the IVP. Note that this is the graph of an approximate solution, which is found with a Runge-Kutta type numerical method.
For what reason are these results so different when you approach 2?
It seems that on the intervals [0, 1], or even [0, 1.5] the approximations are quite good!

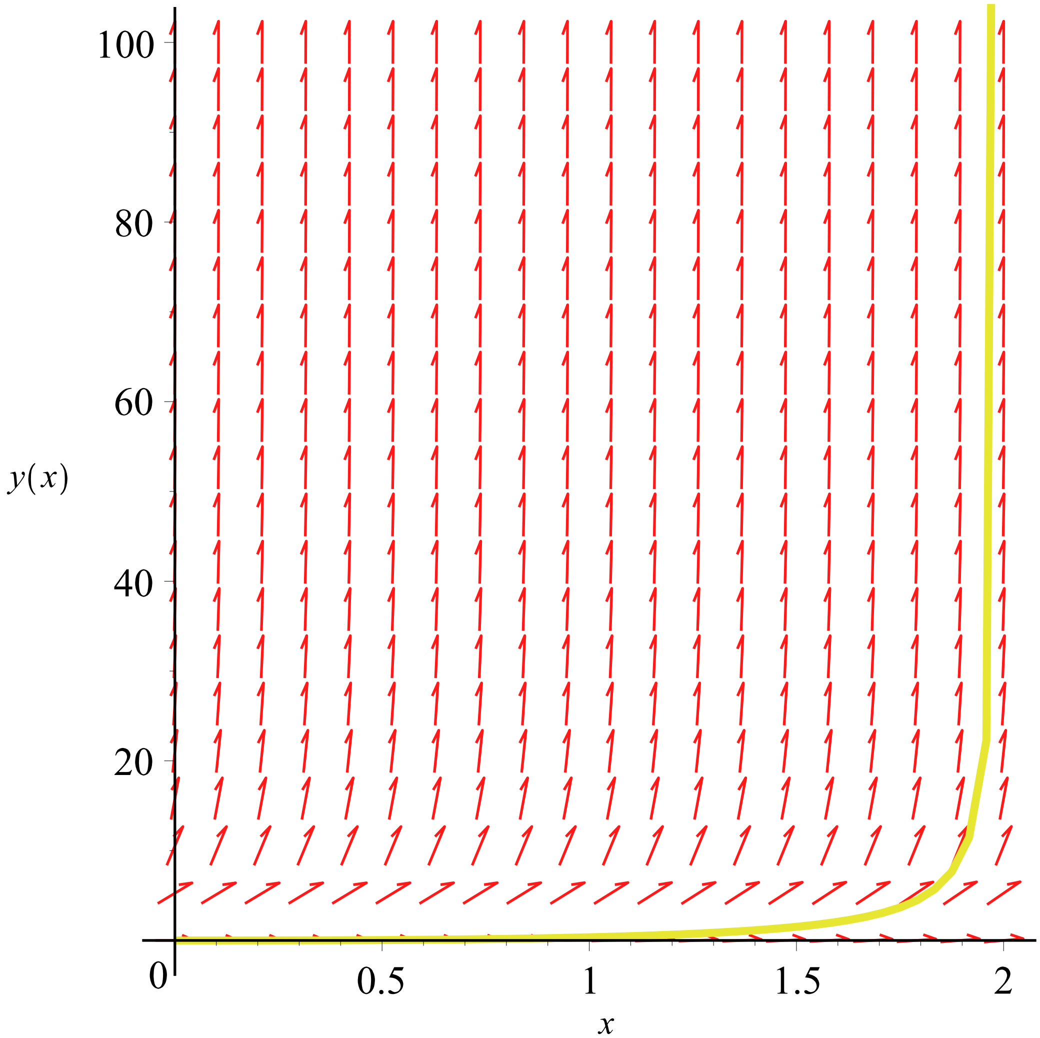
```
> restart: with(DEtools):f:=(x,y)->y^2+x^2; dsolve({diff(y(x),x)=y(x)^2+x^2,y(0)=0});
```

$$f := (x, y) \rightarrow y^2 + x^2$$

$$y(x) = \begin{cases} 0 & x = 0 \\ -\dfrac{\left(\mathrm{BesselJ}\left(-\frac{3}{4}, \frac{1}{2}\, x^2\right) - \mathrm{BesselY}\left(-\frac{3}{4}, \frac{1}{2}\, x^2\right)\right) x}{\mathrm{BesselJ}\left(\frac{1}{4}, \frac{1}{2}\, x^2\right) - \mathrm{BesselY}\left(\frac{1}{4}, \frac{1}{2}\, x^2\right)} & \mathit{otherwise} \end{cases} \quad \textbf{(35)}$$

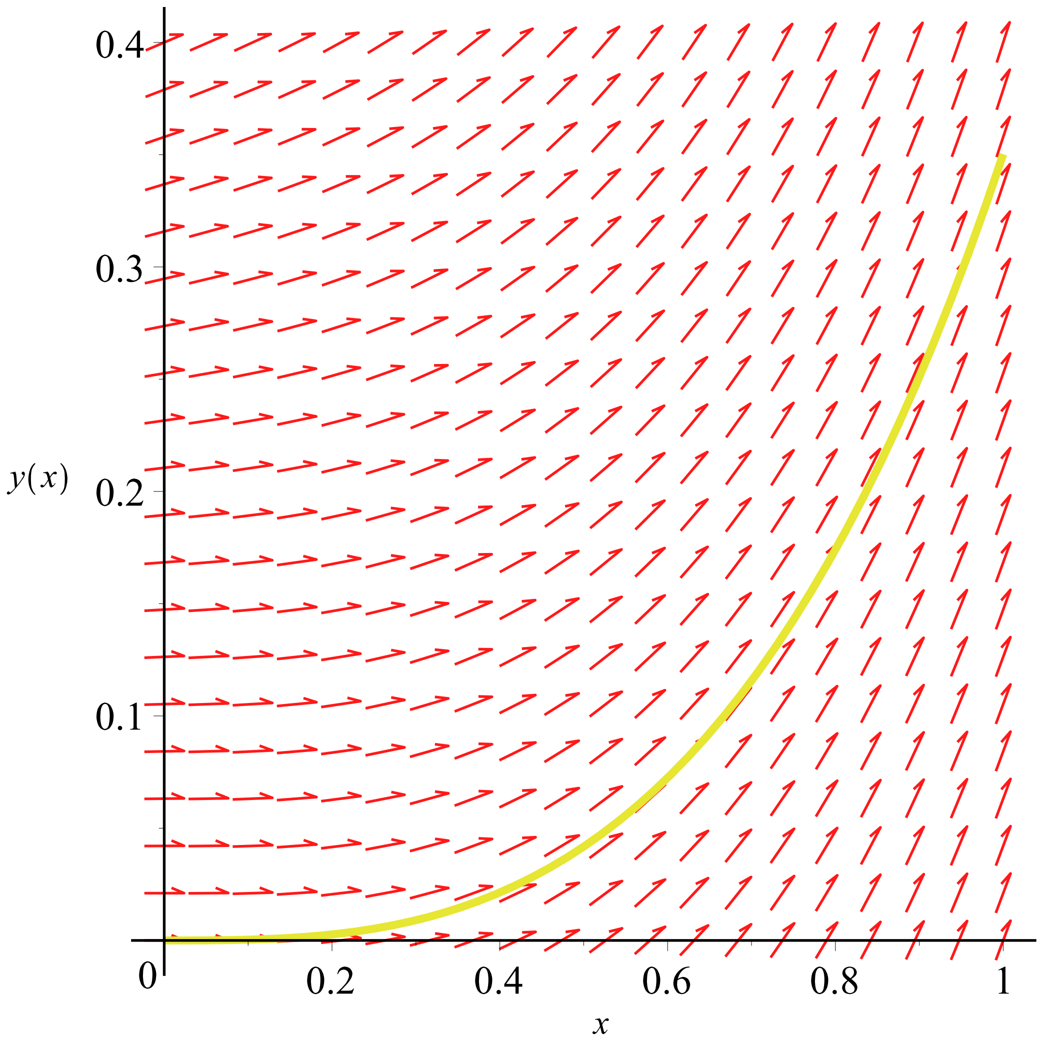
```
> DEplot(diff(y(x),x)=f(x,y(x)),y(x),x=0..2,[[y(0)=0]],y=0..100);
```

It seems that y(2) is greater than 100.




```
> DEplot(diff(y(x),x)=f(x,y(x)),y(x),x=0..1,[[y(0)=0]],y=0..0.4);
```

It seems that y(1) is around 0.35.




```
> h:=0.1; x:=0;y:=0;
```

$$h := 0.1$$
$$x := 0$$
$$y := 0 \tag{36}$$

```
> for i from 1 to 20 do y:=y+h*f(x,y): x:=x+h: print(x,y); od:
```

This method gives that y(1) is 0.2925421046, and y(2) is 5.852099613.

0.1, 0.

0.2, 0.001

0.3, 0.0050001

0.4, 0.01400260010

0.5, 0.03002220738

0.6, 0.05511234067

0.7, 0.09141607768

0.8, 0.1412517676

0.9, 0.2072469738

1.0, 0.2925421046

1.1, 0.4011001929

1.2, 0.5381883294

$$1.3, 0.7111529972$$
$$1.4, 0.9307268557$$
$$1.5, 1.213352104$$
$$1.6, 1.585574437$$
$$1.7, 2.092979066$$
$$1.8, 2.820035203$$
$$1.9, 3.939295058$$
$$2.0, 5.852099613 \qquad (37)$$

```
> h:=0.1; x:=0;y:=0;
```

$$h := 0.1$$
$$x := 0$$
$$y := 0 \qquad (38)$$

```
> for i from 1 to 20 do y:=y+h/2*f(x,y)+h/2*f(x+h,y+h*f(x,y)): x:=
x+h: print(x,y); od:
```

This method gives that y(1) is 0.3518301326, and y(2) is 23.42048639.

$$0.1, 0.0005000000000$$
$$0.2, 0.003000125004$$
$$0.3, 0.009503025760$$
$$0.4, 0.02202467595$$
$$0.5, 0.04262140864$$
$$0.6, 0.07344210066$$
$$0.7, 0.1168165840$$
$$0.8, 0.1753963673$$
$$0.9, 0.2523742135$$
$$1.0, 0.3518301326$$
$$1.1, 0.4792938348$$
$$1.2, 0.6427029949$$
$$1.3, 0.8541363558$$
$$1.4, 1.133184603$$
$$1.5, 1.514119178$$
$$1.6, 2.062972003$$
$$1.7, 2.924894430$$
$$1.8, 4.487143656$$
$$1.9, 8.165117641$$
$$2.0, 23.42048639 \qquad (39)$$

```
>
```

So, we obtain for y(1) the values 0.2925421046 (Euler's method), 0.3518301326 (improved Euler's method), around 0.35 (the method behind DEplot).
Also, we obtain for y(2) the values 5.852099613 (Euler's method), 23.42048639 (improved Euler's method), greater than 100 (the method behind DEplot).

We mention that the Runge-Kutta method behind DEplot is more advanced than the other two. If we look at the direction field and at the approximate graph plotted with DEplot, we see just before the value x=2, a sudden change of direction to the left with almost 90 degrees. This is a very difficult manoeuvre for the first two methods when using the stepsize h=0.1 (maybe it is not small enough). You can try with a smaller stepsize.

Problem **6.** For the IVP y'=-250y, y(0)=1, apply both methods in the interval [0,1] with step size h=0.1. Compare the approximate values of the solution with the exact one. Write the error in each case in your notebook. Have you ever seen such a huge error?

```
> restart; f:=(x,y)->-250*y; sol6:=dsolve({diff(y(x),x)=f(x,y(x)),y
  (0)=1}); phi:=unapply(rhs(sol6),x);
```

$$f := (x, y) \to -250\, y$$

$$sol6 := y(x) = e^{-250\, x}$$

$$\phi := x \to e^{-250\, x} \quad \textbf{(40)}$$

```
> h:=0.1; x:=0;y:=1;
```

$$h := 0.1$$

$$x := 0$$

$$y := 1 \quad \textbf{(41)}$$

```
> for i from 1 to 10 do y:=y+h/2*f(x,y)+h/2*f(x+h,y+h*f(x,y)): x:=
  x+h: print(x,y,phi(x),abs(y-phi(x))); od:
```

The errors are on the last column.

$$0.1,\ 288.5000000,\ 1.388794386\ 10^{-11},\ 288.5000000$$

$$0.2,\ 83232.25000,\ 1.928749848\ 10^{-22},\ 83232.25000$$

$$0.3,\ 2.401250412\ 10^{7},\ 2.678636962\ 10^{-33},\ 2.401250412\ 10^{7}$$

$$0.4,\ 6.927607438\ 10^{9},\ 3.720075976\ 10^{-44},\ 6.927607438\ 10^{9}$$

$$0.5,\ 1.998614746\ 10^{12},\ 5.166420633\ 10^{-55},\ 1.998614746\ 10^{12}$$

$$0.6,\ 5.766003544\ 10^{14},\ 7.175095973\ 10^{-66},\ 5.766003544\ 10^{14}$$

$$0.7,\ 1.663492023\ 10^{17},\ 9.964733010\ 10^{-77},\ 1.663492023\ 10^{17}$$

$$0.8,\ 4.799174487\ 10^{19},\ 1.383896527\ 10^{-87},\ 4.799174487\ 10^{19}$$

$$0.9,\ 1.384561839\ 10^{22},\ 1.921947728\ 10^{-98},\ 1.384561839\ 10^{22}$$

$$1.0,\ 3.994460906\ 10^{24},\ 2.669190216\ 10^{-109},\ 3.994460906\ 10^{24} \quad \textbf{(42)}$$

```
> y;
```

$$3.994460906\ 10^{24} \quad \textbf{(43)}$$

```
> evalf(abs(y-exp(-250)));
```

$$3.994460906\ 10^{24} \quad \textbf{(44)}$$

```
> h:=0.0000001; x:=0;y:=1;
```

$$h := 1.\ 10^{-7}$$

$$x := 0$$

$$y := 1 \quad \textbf{(45)}$$

```
> for i from 1 to 10^7 do y:=y+h/2*f(x,y)+h/2*f(x+h,y+h*f(x,y)):
  x:=x+h:  od:
> y;
```

$$2.669190520\ 10^{-109} \quad \textbf{(46)}$$

```
> evalf(abs(y-exp(-250)));
```

$$\textbf{(47)}$$

$$3.04\ 10^{-116} \tag{47}$$

>

Conclusion: the stepsize must be adapted to each problem. In this case, the error in x=1 when using the stepsize h=0.1 is $3.994460906 \cdot 10^{24}$
while when using the stepsize h=10^(-7), is $3.04 \cdot 10^{-116}$