2015-10-28-082430

Denis Gaidashev

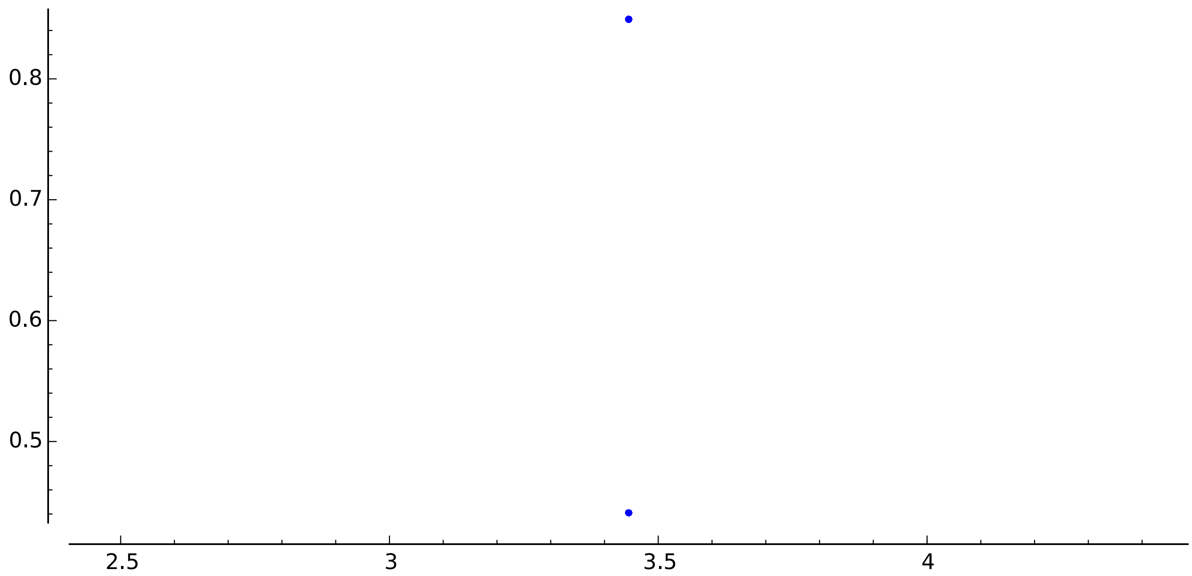
2015-12-20

Contents

```
x,m=var('x,m')

#q and qder are function of x and m
q=function('q',x,m)
qder=function('qder',x,m)
q=m*x*(1-x)
qder=diff(q,x)

#example of evaluation of qder:
qder.subs(x=0.2,m=3)

#iterating q with m=m0, so that iteration would converge to an \
    attracting  cycle
x0=0.564 #some arbitrary initial condition
m0=3.445
j=1;
while j<1000:
    x0old=x0;
    x0=q.subs(x=x0,m=m0);
    j=j+1;


N=100
#point x0 is very close to some periodic point in some cycle. Now I \
    will iterate this point N times and plot all these ierates.

#pts is an array of points (q^j(x0),m0), j<N, the next line \
    initiates it by creating an empty array
pts=[]
j=1;
while j<N:
      x0=q.subs(x=x0,m=m0);
```

```
    pts.append((m0,x0)) # here we append the array by adding \
   another point
    j=j+1;

#command p=point(pts) creates a graphic object which consists of N \
   points.
p=point(pts)
#inputing p simply plots this graphic object
p
#the result are points in an attractive cycle ploted vertically \
   above the parameter m0.

#the plot seems to be a period two point. Lets compute the \
   derivative of q^k(x0), k=2 in this case, using the chain rule
j=1;
k=2;
d=1;
while j<=k:
    x1=qder.subs(x=x0,m=m0);
    x0=q.subs(x=x0,m=m0);
    d=d*x1;
    j=j+1;

d #shows the derivative, the value for this parameter is close to \
   the bifurcation value -1

#Now to create a bifurcation diagram you need to loop over \
   parameters
```

```
1.80000000000000
```




```
-0.978024618437272
```