TUTORIAL MI:
Iteration of maps and continuation of fixed points and cycles in MatcontM

May 3, 2016

This tutorial shows how to use MATCONTM to compute orbits of maps

$$u \mapsto f(u, \alpha), \quad u \in \mathbb{R}^n, \alpha \in \mathbb{R}^m,$$

at fixed parameter values, and to continue fixed points and periodic orbits (cycles) in one parameter.

We study the following recurrence equation (Maynard Smith 1968):

$$x_{k+1} = r x_k (1 - x_{k-1}) + \epsilon$$

where x_k is the density of a population at year k, r is the growth rate, and ϵ is the migration rate. It is assumed that the growth is determined not only by the current population density but also by its density one year before. Introduce $y_k = x_{k-1}$. The population dynamics can be then studied by iterating the planar map

$$\begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} r x (1 - y) + \epsilon \\ x \end{pmatrix}.$$

This map is called the *delayed logistic map.*

1 Installation and system specification

Obtain the latest file `MatcontM_20xx.xx.xx.zip`[1]. Extract `MatcontM_20xx.xx.xx.zip` and change the working directory of MATLAB to the root of the extracted folder. You should be able to see the file `matcontm.m`. You start MatcontM by executing in MATLAB:

```
>> matcontm
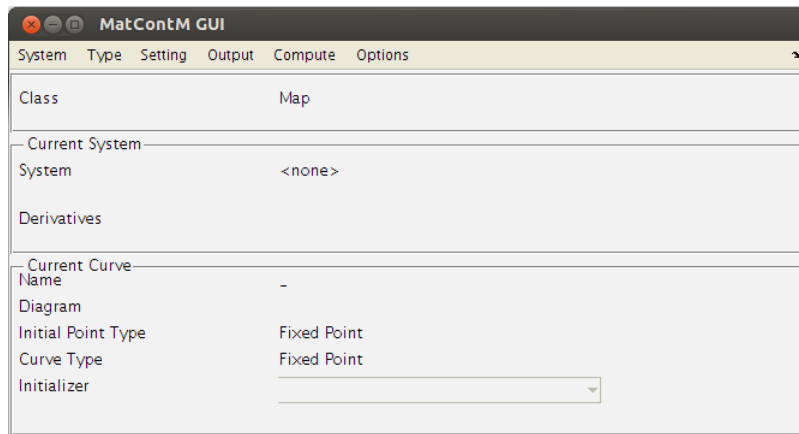```

This opens the main MatcontM window (Figure 1).



Figure 1: Main Window of MatcontM (empty)

---

Select the menu item **System** in the upper left corner and then select **System-browser** (Figure 2). In what follows, we will use the notation **System|Systembrowser**. A Data Browser window appears showing the list of available systems (maps). The right panel contains a summary of the highlighted system in the list. Underneath the list is a series of buttons that perform system-related actions. Press the `New`-button to add a new system. An empty System window appears.
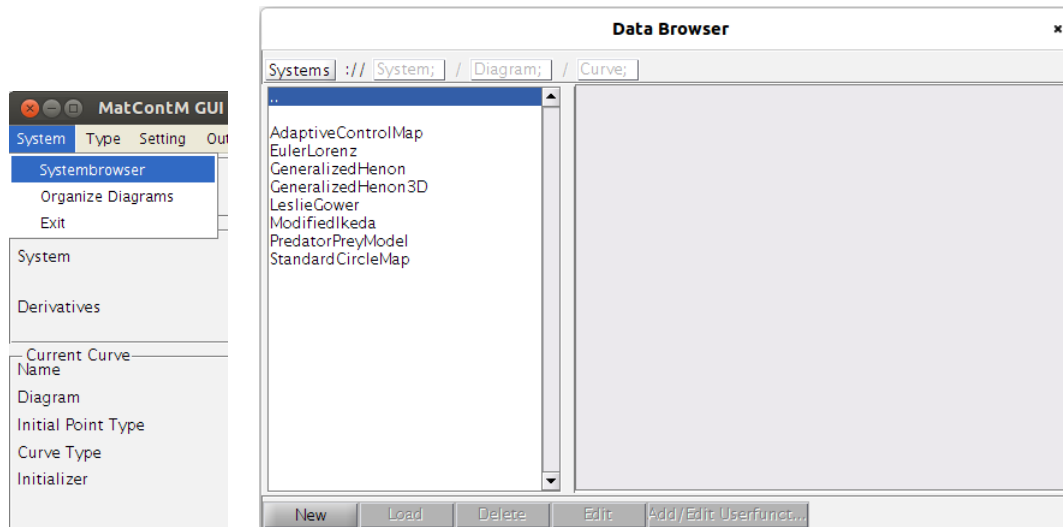


Figure 2: The System browser displays the list of systems present in MatcontM with a summary of each system. The lower left buttons are for creating, loading, deleting and editing systems.

Specify the delayed logistic map by entering the data shown in Figure 3. Fill in the name, coordinates and parameters of the system. When adding your own system, make sure that your entry follows this format strictly. No spaces are allowed between the coordinates and between the parameters. When entering the equations, make sure there are no spaces between the '=' symbol and the coordinate, enter "x'=..." not "x'=...". In our case, we have requested that the first 3 order derivatives are symbolically calculated in advance by the MATLAB symbolic toolbox.

Press **OK**. The adding of a system can take a few seconds. This map is now configured and selected for further use. After configuration, the default Starter and Continuer windows should now appear.
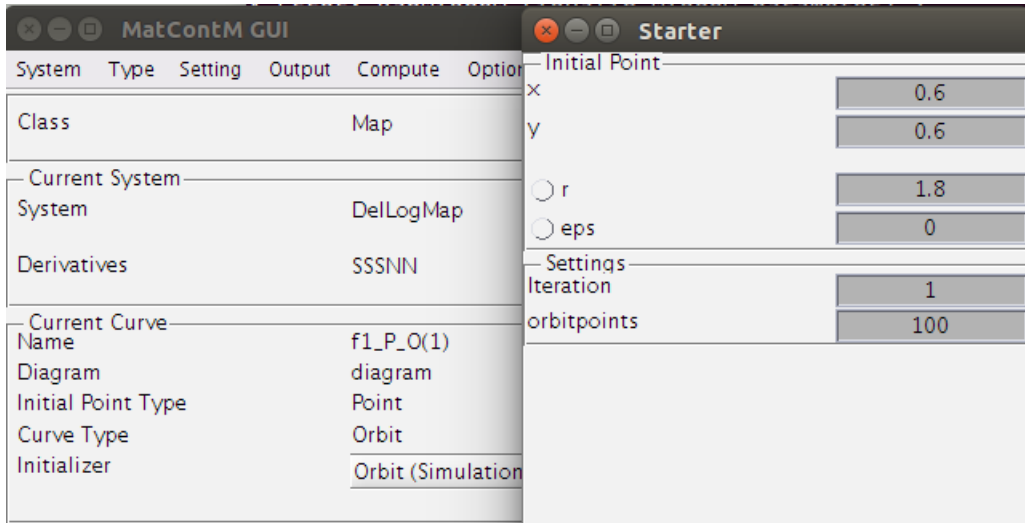
Figure 3: The System input window.

Figure 4: Main window and a configured Starter window.

# 2 Orbit iteration

Select **Type|Initial Point|Point** in the main window (Figure 1). Select the Starter window [2] and set initial data for iterations (Figure 4):

|      |     |
|------|-----|
| x    | 0.6 |
| y    | 0.6 |
| r    | 1.8 |
| eps  | 0   |

Select **Output|Graphic|2D plot** to open a graphic window (see Figure 5). You can change attributes of the open Plot2D window by selecting the **MatcontM|Layout** menu in the upper-right corner. No changes to the default layout are needed in this example [3].

---

[2]You can reopen this window by selecting **Setting|Starter** in the main window.

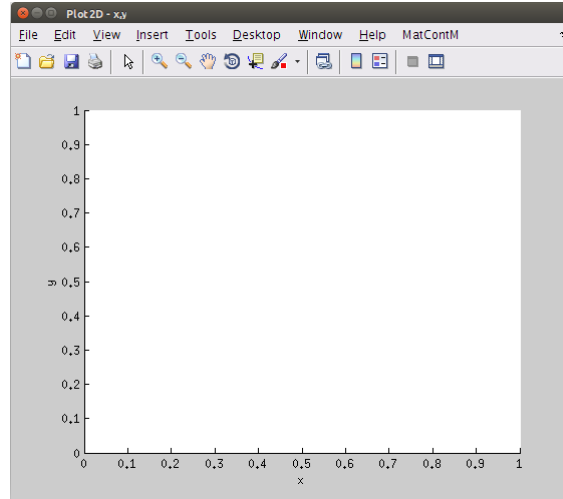[3]The menu **MatcontM|Orbit Options** in the Plot2D window allows you to add point numbers and connecting lines.

Figure 5: The default Plot2D window.

Select **Compute|Forward** in the main window[4]. The Output window will appear (can be seen in Figure 8, left). This window allows you to pause, stop and resume computation. This window also displays information during computation. For now, you only need to close this window after computation is finished.

The computations produce an orbit converging to a fixed point (Figure 6, left). Clear the figure by selecting **MatcontM|Clear**. Increase the number of computed points by setting the value of `orbitpoints` to 300 in the Starter window. Increase now the value of $r$ to 2.1 and iterate the map with **Compute|Forward**. The orbit now approaches a *closed invariant curve* (Figure 6).
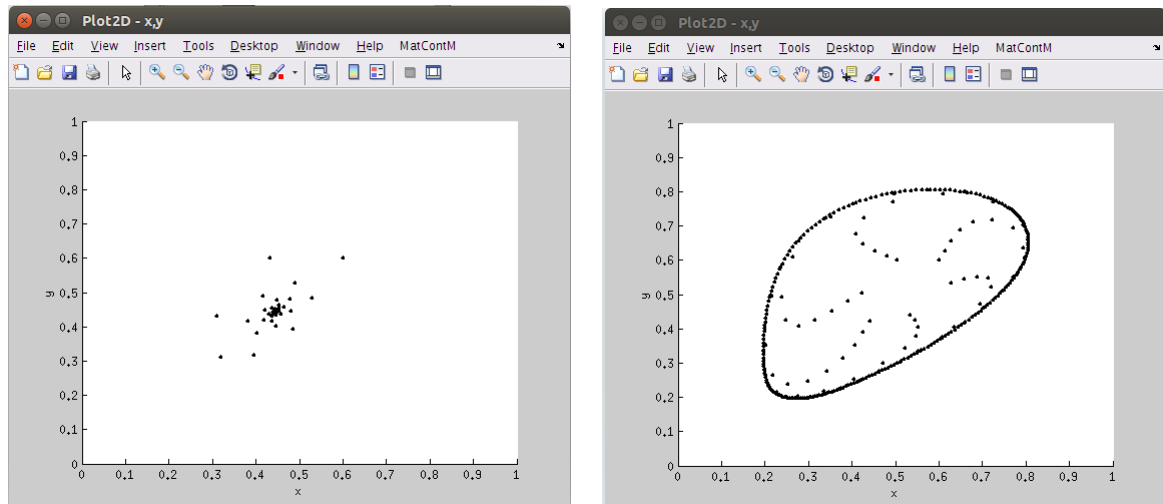


Figure 6: Convergence to a fixed point (left) and to a closed invariant curve (right).

---

[4]Backward iteration is not supported.

# 3 Fixed point continuation

Change the value of $r$ back to 1.8 and recompute the orbit by selecting **Compute|Forward**. Select the **View Curve** button in the Output window to browse the latest computed curve. You can also obtain this window by right-clicking in the Current Curve section of the main window. Scroll down and highlight the last computed point (see Figure 7). Double-click or press the low right button **Select Point**. The values for $x$ and $y$ in the Starter window should now be close to 0.44444 .
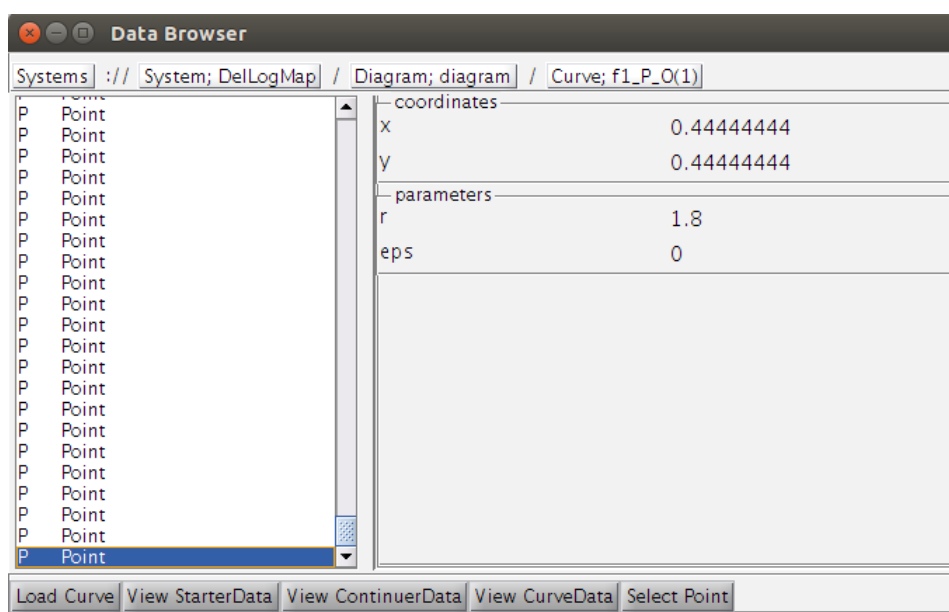


Figure 7: The last point is to be selected.

Select **Type|Initial Point|Fixed Point**. A fixed point continuation is now selected. Activate the parameter $r$ in the Starter window by clicking the radio button. Activate the Calculate multipliers option to monitor the eigenvalues (multipliers). Open the Numeric window by selecting **Output|Numeric** in the main window. In the same window, select **Options|Pause** and opt to suspend computation `At Each Point`, and press **OK**. Clear the Plot2D window.

Select **Compute|Forward**. Press the **Resume** button in the Output window to compute a new point. You will be able to check that the fixed points become unstable as two multipliers cross the unit circle at $r = 2.0$ (Figure 8). This is a supercritical Neimark-Sacker (NS) bifurcation (normal form coefficient of NS is $-1.0$) that gives rise to the stable closed invariant curve that we observed already. Notice that the moduli of the multipliers are close to one, while their arguments $\theta_{1,2} \approx \pm 60°$. Press the **Stop** button after detecting the Neimark-Sacker bifurcation and close the Output window. We permanently save the curve by giving it a unique name by right-clicking on the Current Curve panel in the main window. Select **Rename Curve** and enter e.g. '`Fixed Point(+)`'.
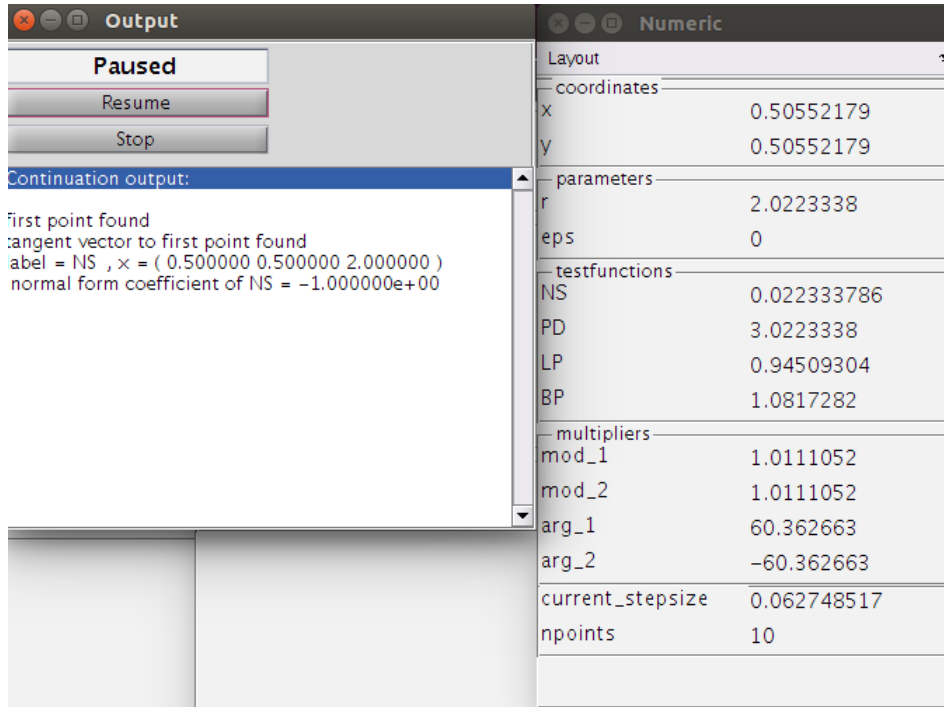
Figure 8: The Output window and the Numeric window during the computation of a fixed point continuation. A supercritical Neimark-Sacker bifurcation was just detected.

# 4 Phase locking

Return now to simulations by selecting **Type|Initial Point|Point** but inputing $r = 2.177$ in the Starter window. Make sure Orbit (Simulation) is selected as Initializer in the main window (by selecting **Type|Curve|Orbit (Simulation)**). Clear the Plot2D window.

The iterations with **Compute|Forward** (which you have to resume at each point) will result in a stable cycle of period seven! This is the so called phase locking: a quasi-periodic motion on the invariant curve is replaced by purely periodic motion.

Select a point on this cycle with maximal x-value ($x \approx 0.90$) using the data browser. For this, you can open the browser by pressing **View Curve** button. Click then a Point in the end of the list and see the corresponding numerical values in the right panel of the Data Browser window. When a suitable point is found, press **Select Point** button and close the Output window.

We will continue this period-7 cycle as a fixed point of the 7th-iterate of the map with respect to the parameter $r$ by selecting **Type|Initial Point|Fixed Point**. In the Starter window, activate the parameter **r** and set the value of Iteration to 7 (Figure 9, left).
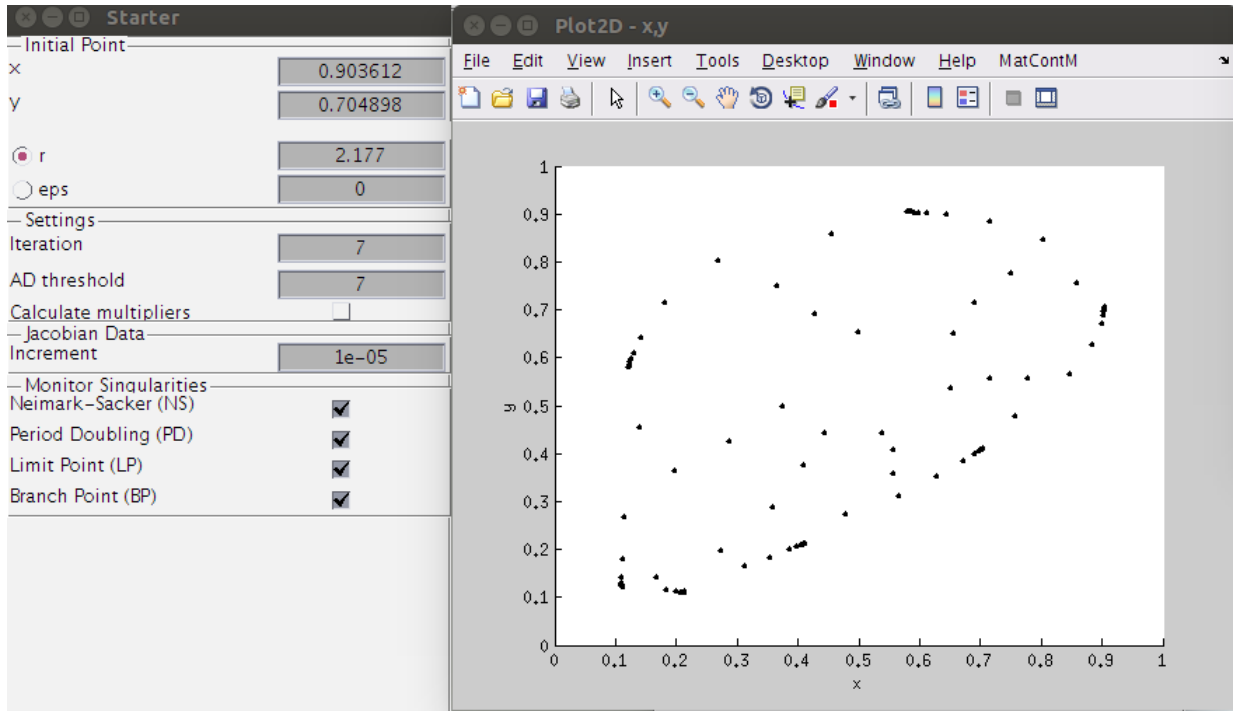
Figure 9: Starter window (left) for the continuation of the fixed point of the 7th iterate (right).

Select the Continuer window [5] and set MaxStepsize to 0.01. It is also useful to change attributes of the 2D Graphic window. Select **MatcontM|Layout** and select parameter r and coordinate x as Abscissa and Ordinate, respectively. It is also advised to change the visibility range for r as seen in Figure 10. Make sure to clear the window using **MatcontM|Clear**.

Before we start computations, it is advised to set **Options|Pause** to Suspend Computation At special points. Select **Compute|Forward**. You need to press **Resume** several times at detected *limit points* (labled with LP) to get a closed curve presented in Figure 11. Note: The upper left LP might not be detected during the computation of the closed curve. This bifurcation is too close to the starting point.

The seventh-iterate of the map has seven stable fixed points and seven saddle fixed points that collide and disappear simultaneously at two limit point (fold) bifurcations, labeled with LP. Thus, the phase locking happens in a narrow parameter window between $r_1 = 2.174...$ and $r_2 = 2.201...$. Store the computed fixed point curve by renaming it (right-click in the Current Curve in the main window) to e.g.7-Fixed point(+).

---

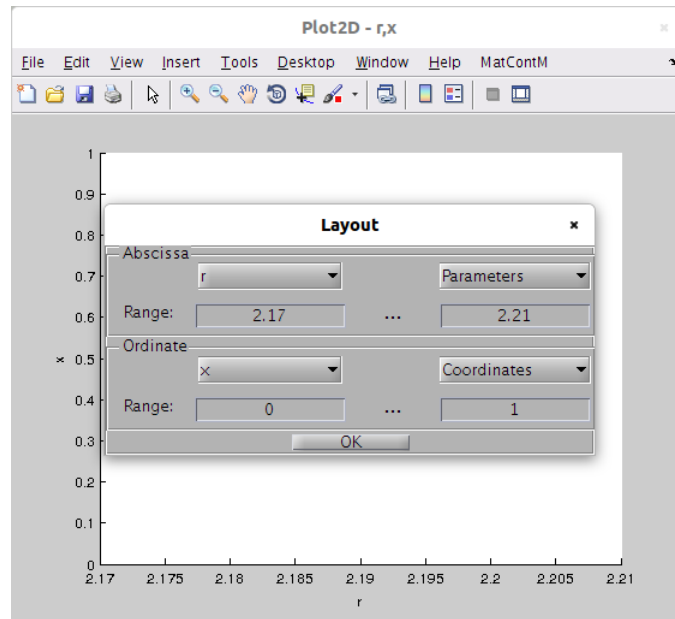[5]You can reopen this window by selecting **Setting|Continuer** in the main window.

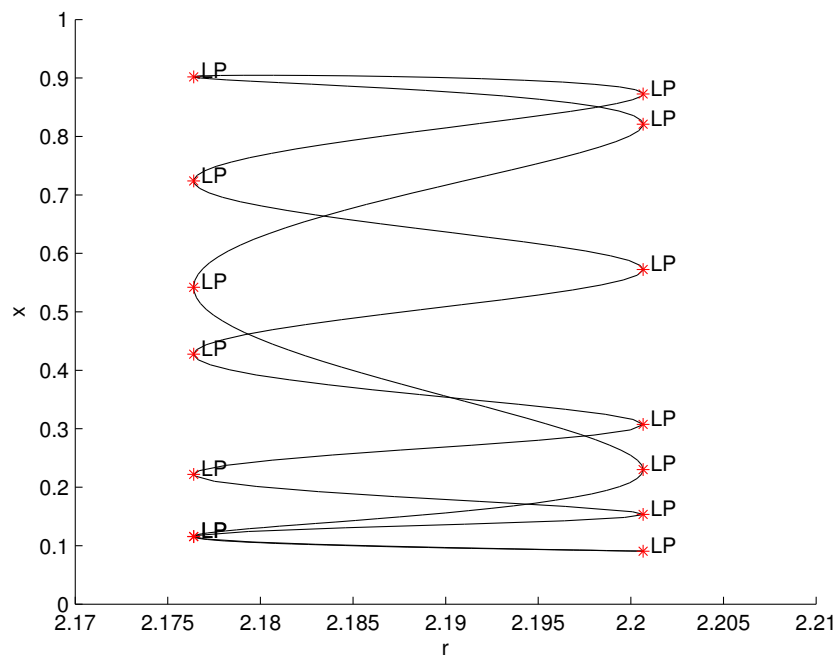Figure 10: Layout window with the new attributes of Plot2D window.



Figure 11: The continuation of the period-7 cycle as the fixed point of 7th iterate of the map.

10

# 5   PD Cascade

One can compute a parameter value $r$ where the period-7 cycle exhibits a period-doubling bifurcation. For this, right-click on the `Current Curve` panel in the main window to display a context menu (see Figure 12), where you select **View Diagram**.
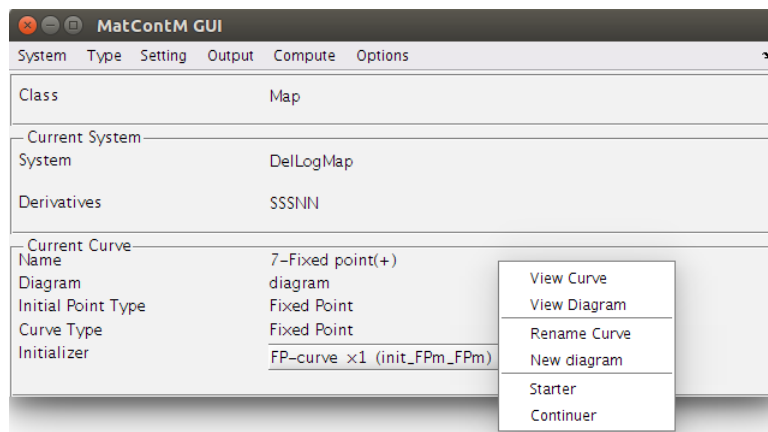


Figure 12: Context menu.

Then select the `7-Fixed point(+)` curve via the Diagram Browser. Highlight the `7-Fixed point(+)` curve and press the **Load** button below. This makes this curve current, as is visible in the main window. Go to the Starter window and deactivate parameter `r` but activate instead the parameter `eps`. Clear the Plot2D window (or create a new one) and change the visibility attributes as suggested in Figure 13.



Figure 13: New visibility attributes.

We will create a new diagram by selecting `New diagram` in the menu shown in Figure 12. Enter the name "`pdcascade`". The current diagram is now set to `pdcascade` instead of the default `diagram`. Curves will now be stored in this new diagram.
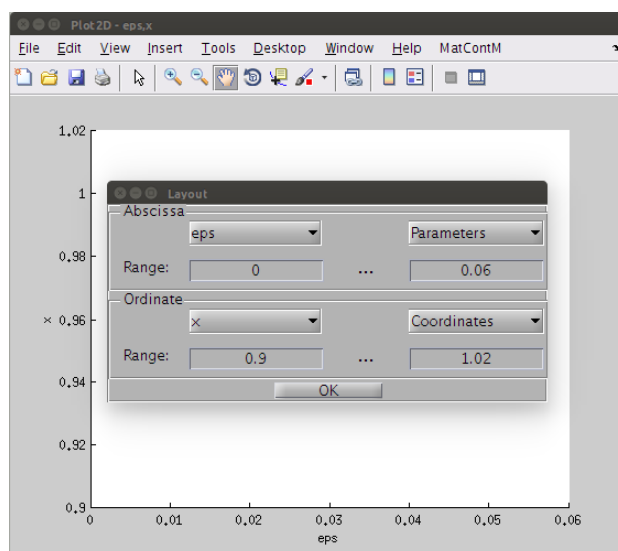
Select **Compute|Forward** until a period-doubling (flip) bifurcation (labeled PD) is detected (Figure 14).
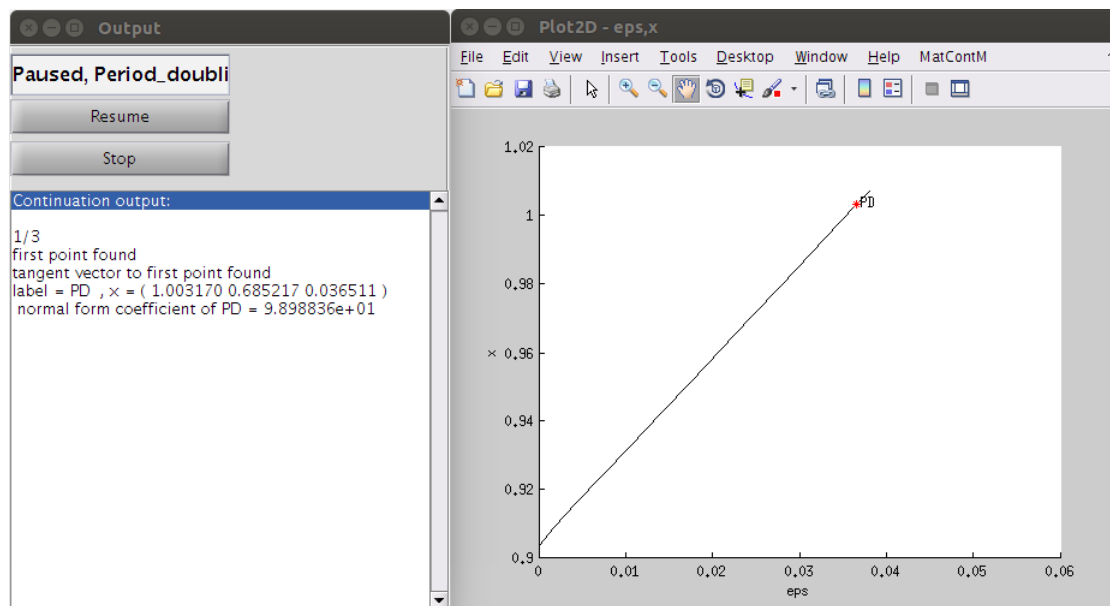


Figure 14: Continuation of the period-14 cycle: its superctitical PD bifurcation is detected.

Once we detect a PD bifurcation, we stop the computations by pressing **Stop**. The normal form coefficient of PD is positive, so we have detected a supercritical period-doubling bifurcation. Select this bifurcation as the initial point using the data browser. We will compute the period-doubled cycle (actually, the 14-cycle for the delayed logistic map). We have to change the `initializer`, since two-parameter continuation of the PD bifurcation curve is selected by default. Change the `initializer` to `FP-curve x2` (see Figure 15). This allows to start the continuation of the bifurcating period-doubled cycle in parameter `eps`. Select **Compute|Forward**, you should obtain Figure 16. Stop after detecting the PD point.

We repeat the process. Select the latest period-doubling bifurcation. You can select bifurcations directly from plot windows by double-clicking on the labels, see Figure 17. A small window will appear which allows you to confirm your selection. Make sure no other MATLAB plot tools are enabled (Zoom, Pan, ...) and all computations are terminanted.

Ensure that the `initializer` is set to `FP-curve x2` and select **Compute|Forward**. This computation will also detect some LP bifurcations, you may stop after detecting the first PD bifurcation.

You can repeat this process once more. However, because the curves are getting too close to each other, we need to sharpen some parameters. Set `amplitude` to 0.0001 in
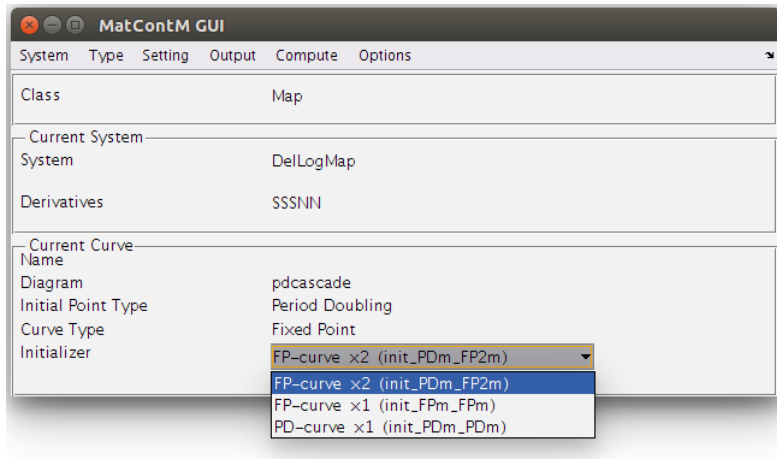
Figure 15: Selection of the initializer from `PD-curve` to `FP-curve x2`.
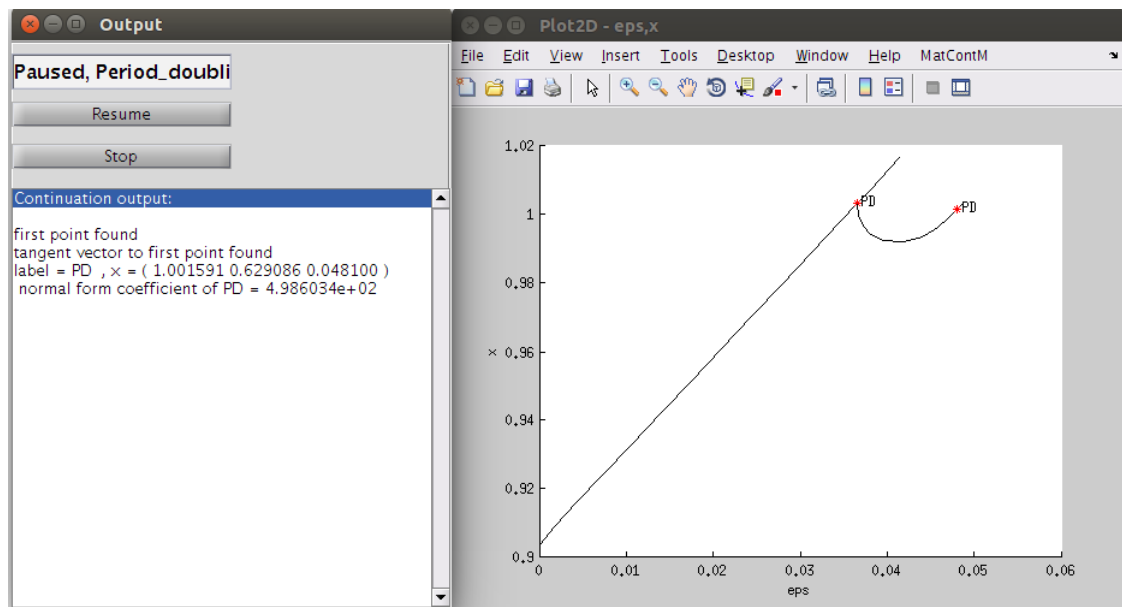


Figure 16: A super-critical period-doubling bifurcation on the doubled-curve (14-cycle) is detected.
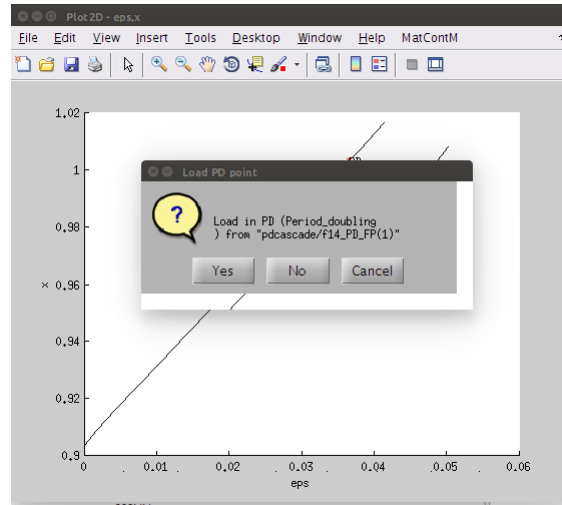
Figure 17

the `Starter` window and set `InitStepSize` and `MaxStepSize` to 0.001 in the `Continuer`
window (Figure 18). The resulting cascade is shown in Figure 19.

# 6    Viewing and Managing Data

Right-clicking on the `Current Curve` panel opens the menu shown in Figure 12. Select
**View diagram**.

The appearing `Data Browser` (Figure 20) gives you an overview of the curves in the
diagram. Use the list on the left to highlight a curve. Selecting **View** will open a window
showing the raw data of the curve. By double-clicking on **..** you will go one level up and
see an overview of the available diagrams. By double-clicking on a curve name you will
go one level down and see the content of the curve as seen in Figure 7. Double-clicking
on a point will select that point as a starting point. The **Load** button shown in Figure 20
and Figure 7 will select the curve in the main window and will load the initial conditions
of the curve in the `Starter` and `Continuer` window. You can change the current diagram
by selecting a curve in another diagram. The diagram will change automatically.

Selecting **MatcontM|Redraw Diagram** in the plot window will redraw the entire
diagram. Sometimes the diagram can be littered with redundant curves. There is a tool
available for moving and remove curves from diagrams.

You can organize diagrams by selecting **System|Organize Diagram** (Figure 21).
This tool allows you to create new diagrams, move curves between diagrams and remove
curves. You can right-click on the panels to obtain a list of actions. Curves can be
removed by right-clicking and selecting the remove action. You can move curves between
two diagrams by first selecting two diagrams using the lists at the top, left and right.
You can move curves around by selecting them in one list and moving them using the
arrow buttons in the middle. At each moment you can redraw the current diagram via

14

| Output | Continuer | Starter |
|---|---|---|
| **Finished** | | |
| View Curve | | |
| Close | | |

Continuation output:

first point found
tangent vector to first point found
label = PD , x = ( 0.994247 0.605505 0.052349 )
 normal form coefficient of PD = 8.858781e+04
elapsed time = 7.9 secs
npoints curve = 9

**Continuer**

— Continuation Data—
| InitStepSize | 0.001 |
| MinStepSize | 1e-05 |
| MaxStepSize | 0.001 |

— Corrector Data—
| MaxNewtonIters | 3 |
| MaxCorrIters | 10 |
| TestIters | 10 |
| VarTolerance | 1e-06 |
| FunTolerance | 1e-06 |
| TestTolerance | 1e-05 |
| Adapt | 0 |

— Stop Data—
| MaxNumPoints | 300 |
| ClosedCurve | 50 |

**Starter**

— Initial Point—
| x | 0.997454 |
| y | 0.610132 |
| ○ r | 2.177 |
| ◉ eps | 0.0522493 |

— Settings—
| Iteration | 28 |
| AD threshold | 7 |
| Amplitude | 0.0001 |
| Calculate multipliers | ☐ |

— Jacobian Data—
| Increment | 1e-05 |

— Monitor Singularities—
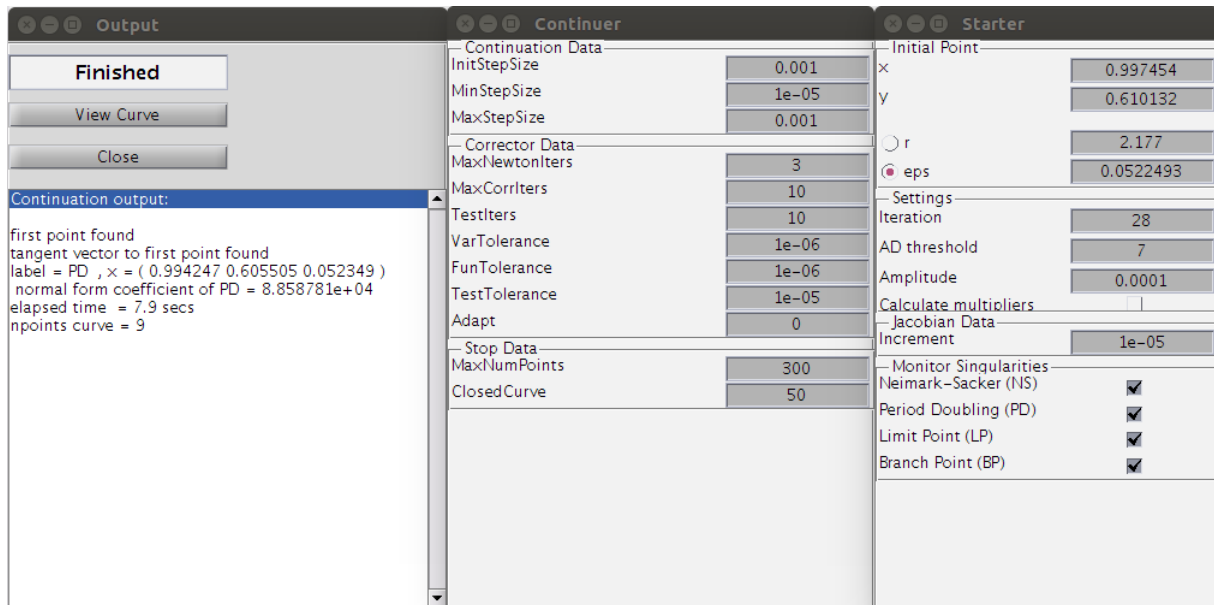| Neimark–Sacker (NS) | ☑ |
| Period Doubling (PD) | ☑ |
| Limit Point (LP) | ☑ |
| Branch Point (BP) | ☑ |

Figure 18: Computing the period-doubling cascade gets increasingly hard because the curves lie too close to each other. We can find the fourth period-doubling bifurcation (See Output window) on a 28-cycle using this setting.
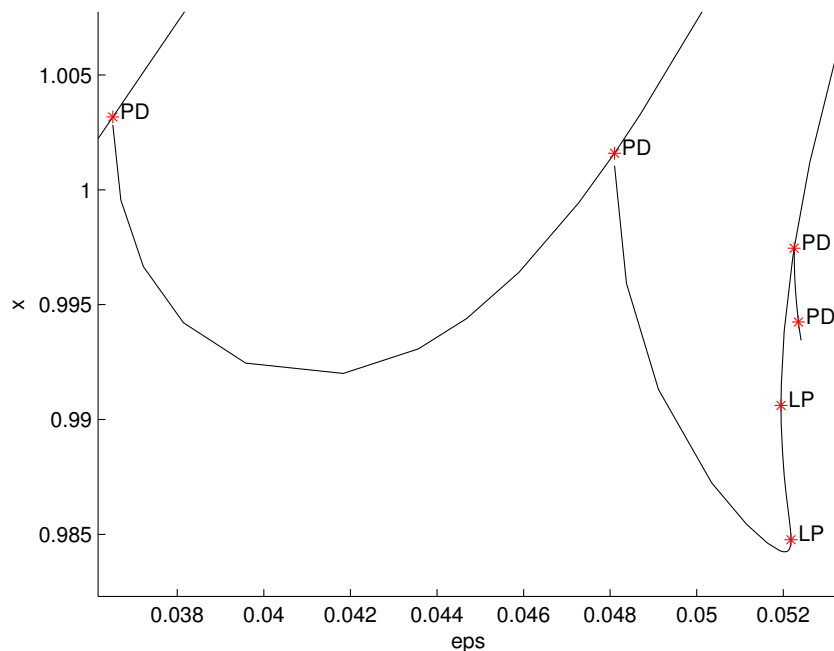


Figure 19: A zoom in the period-doubling cascade. The left curve is the original 7-cycle. The period doubles when a curve emanates from a PD bifurcation.
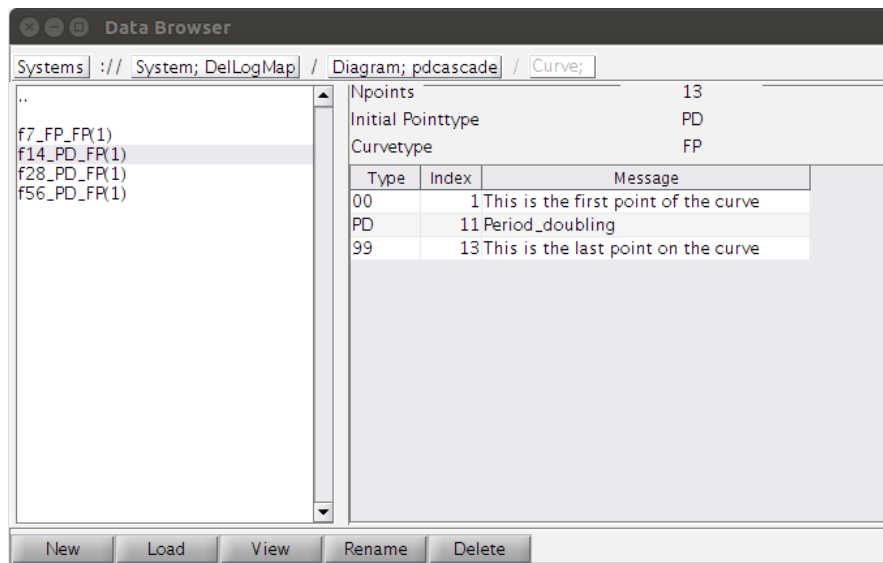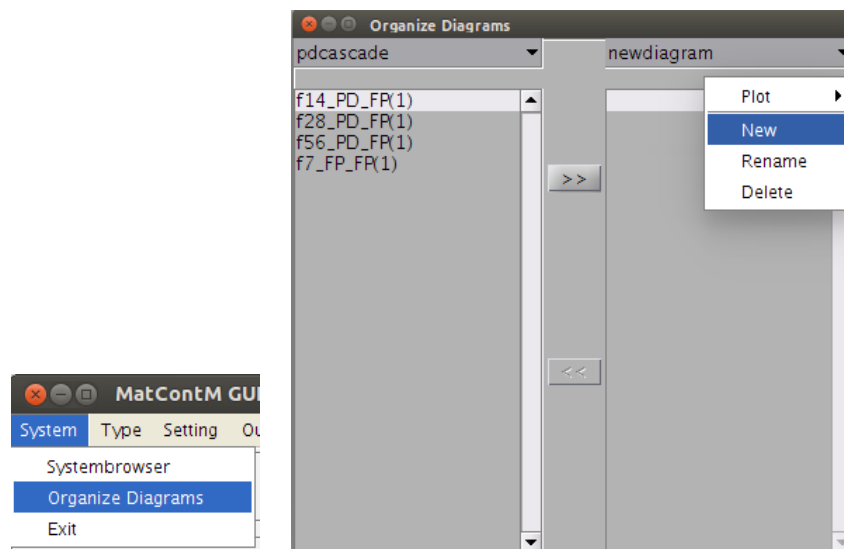
Figure 20: Diagram data browser.



Figure 21: Diagrams `pdcascade` and `newdiagram` are selected for exchange. A curve context menu is visible.

**MatcontM|Redraw Diagram** in the plot window.

The raw data can easily be exported to the MATLAB Command Window. You need to create an interface and retrieve the data.

```
>> cli = CommandLineInterface        %create interface
>> data = cli.getCurve               %retrieve current curve data as struct
data =
    x: [3x13 double]
    v: [3x13 double]
    s: [3x1 struct]
    h: [6x13 double]
    f: [NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN]

>> data.x
ans =
  Columns 1 through 11

    1.0028    0.9996    0.9966    0.9942    0.9925    0.9920    ...
    0.6843    0.6748    0.6653    0.6556    0.6459    0.6361    ...
    0.0365    0.0367    0.0372    0.0381    0.0396    0.0418    ...

  Columns 12 through 13

    1.0033    1.0081
    0.6297    0.6319
    0.0487    0.0502
```

The `getCurve` command will load the data of the curve selected in the main window. The 14-cycle was selected.

7 Additional Problems

A. **Conservative Dynamics** Simulate the following *area-preserving* 2D-map (Hénon 1969)

$$\begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} x\cos\alpha - (y - x^2)\sin\alpha \\ x\sin\alpha + (y - x^2)\cos\alpha \end{pmatrix}$$

for $\cos\alpha = 0.8, 0.4, 0.24, 0.02, 0$ in the domain $(x, y) \in [-1, 1] \times [-1, 1]$. Pay special attention to fixed points and cycles.

B. **Feigenbaum Cascade in Ricker Model** Consider the following simple population model

$$x_{k+1} = \alpha x_k e^{-x_k}$$

where $x_k \geq 0$ is the population density in year k, and $\alpha \geq 0$ is the growth rate. Study the fixed points and cycles of the corresponding scalar map

$$x \mapsto \alpha x e^{-x}.$$

1. Starting from $x = 0$ at $\alpha = 0$, compute with MATCONTM a branch of trivial fixed point of this map in the (x, α)-space. Detect a branching point, and switch to the non-trivial branch of the fixed points using an appropriate `Initializer` option.

2. Detect a period-doubling (PD) bifurcation of this fixed point and switch to the continuation of the bifurcating period-2 cycle.

3. Find several further PD bifurcations and observe numerically the accumulation of the period doubling bifurcations.

4. Study analytically the first PD bifurcation, i.e. find the critical parameter value and compute the PD normal form coefficient. Do they agree with your numerical observations ?