

TUTORIAL I:
Using MATCONT for numerical integration of ODEs

Yu.A. Kuznetsov
Department of Mathematics
Utrecht University
Budapestlaan 6
3508 TA, Utrecht

N. Neiryck
Department of Applied Mathematics,
Computer Science and Statistics
Ghent University
Krijgslaan 281, S9
B-9000, Ghent

October 26, 2020

This session was tested on MATCONT7.0 with MATLAB2018b. It illustrates how to input a system of autonomous ordinary differential equations (ODEs)

$$\dot{x} = f(x, \alpha), \quad x \in \mathbb{R}^n, \alpha \in \mathbb{R}^m,$$

into MATCONT and numerically integrate it with simultaneously visualizing orbits in graphic windows.

It is not possible to load into MATCONT a system without any parameters. However, you can introduce a dummy variable to perform time-integration.

We will study the *Rössler chaotic system*:

$$\begin{cases} \dot{x} &= -y - z \\ \dot{y} &= x + Ay \\ \dot{z} &= Bx - Cz + xz, \end{cases}$$

where (x, y, z) are the phase variables, and (A, B, C) are the parameters.

1 Getting started

We assume that MATCONT is placed into the directory MATCONT of your system and has been installed properly¹. Start MATLAB and change the current directory to MATCONT. Start MATCONT by typing

¹With this we refer to the compilation of the C-files in the subdirectory `LimitCycle`; for details see documentation at <https://sourceforge.net/projects/matcont/>

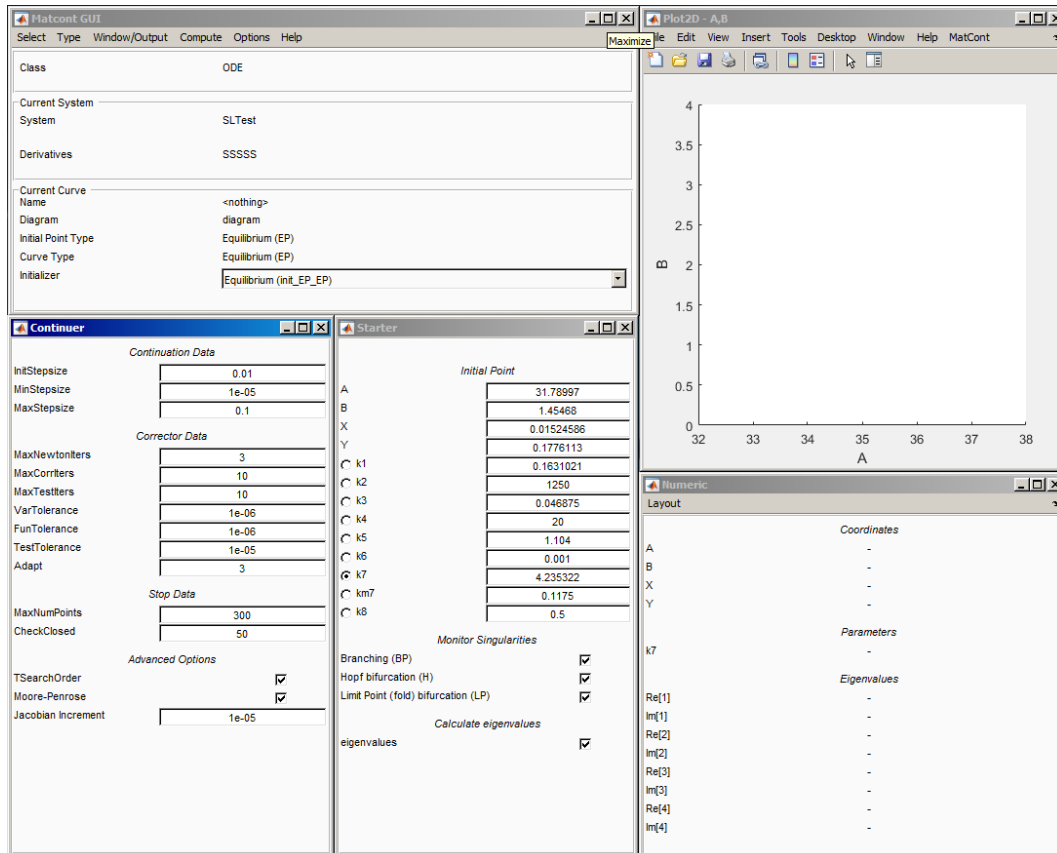


Figure 1: A typical MATCONT startup screen.

`matcont`

in the MATLAB command line window and press Enter.

You will get several windows related to a default ODE system, like in Figure 1. The position of the windows might be rearranged if they do not fit the screen.

The main window is called **MatCont** and has several menus. For instance, to end your MATCONT session, choose the **Exit** item in the **Select** menu. Hereafter this operation will be indicated with **Select|Exit**.

Note. It is sometimes desirable not to load a previously used system (e.g. because the last used system was in some way corrupted). Then by typing

```
matcont clean
```

one gets a fresh start with a blank main MATCONT window.

2 Input new system

The command **Select|System|New** opens the **System** window, which contains several fields and buttons. To identify the system, type for example

```
Rossler
```

in the **Name** field (it must be one word).

Input names of the **Coordinates**: x, y, z , the **Parameters**: A, B, C , and the name for **Time**: t (default).

If shown, select symbolic generation of the 1st order derivatives by pressing the corresponding radio-button ².

Finally, in the large input field, type the RHS of Rössler's system as

```
x'=-y-z
y'=x+A*y
z'=B*x-C*z+x*z
```

Avoid typical mistakes:

- Make sure the multiplication is written explicitly with $*$.
- Specify differential equations in the same order as the coordinates.

Now the **System** window should look like in Figure 2, and you can press **OK** button.

If you made no typing mistakes, the **System** window disappears, and you will see in the main MATCONT window that **Rossler** becomes the current **System** of MATCONT. If selected, the information field **Derivatives** shows the string **SNNN** meaning that the symbolic 1st order derivatives of the RHS will be used.

If you want to change or correct an existing system, click **Select|Systems|Edit/Load/Delete**, select this system in the appearing **Systems** window, and press **Edit** button. The inputted system can be completely erased by selecting **Action|Delete** there.

3 Selection of solution type

To tell MATCONT that we want to integrate the system, i.e. to numerically solve the initial value problem for it, we have to specify the type of the initial point and the type of the curve to compute. To select the initial point type, input **Type|Initial Point|Point**, which means that the initial point has no special properties. To select the curve type, click **Type|Curve|Orbit**³.

²If the MATLAB Symbolic Toolbox is present, there will be buttons indicated 'symbolically'. The first-order derivatives are used in some of the integration algorithms, the first- and second-order derivatives are used in the continuation, while the third-order derivatives are employed in the normal form computations.

³Actually, it is default.

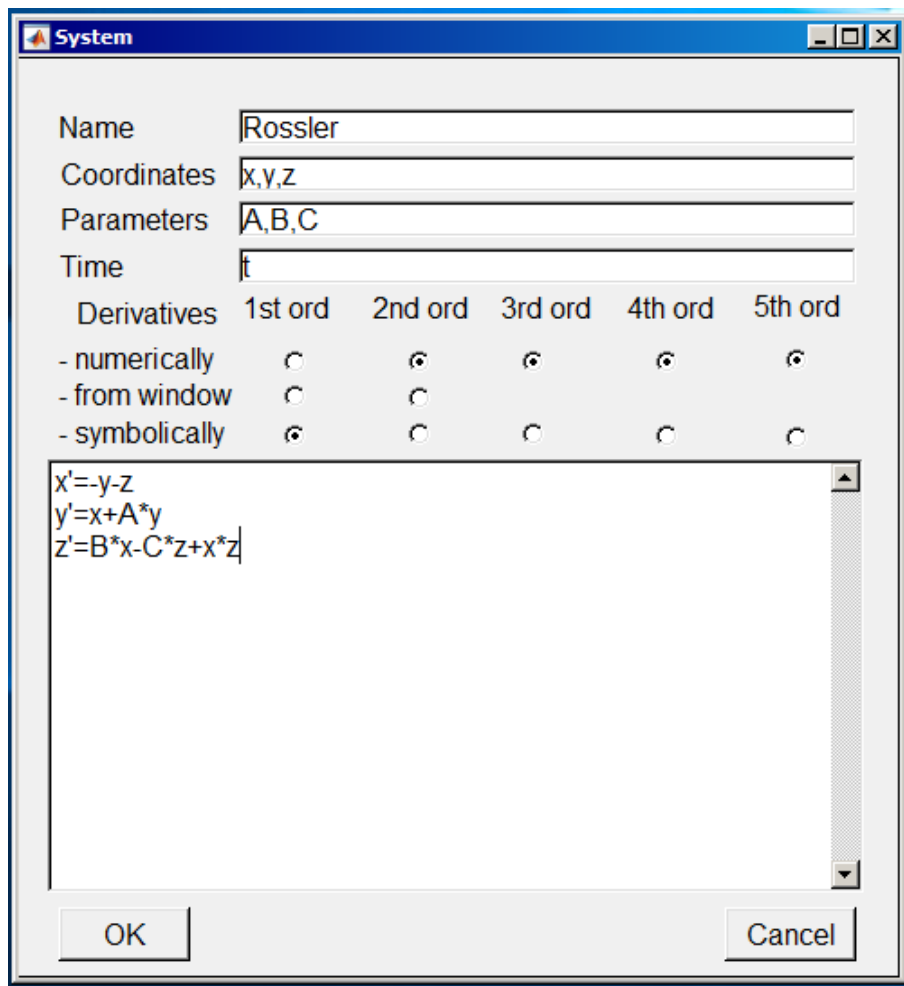


Figure 2: Specifying a new model.

The information fields in the MATCONT main window will reflect the selections, and two more windows appear. These are the corresponding **Starter** and **Integrator** windows. Move them, if necessary, to make both visible.

4 Setting initial data for integration

The **Starter** window is curve-dependent and is used here to specify initial data for the integration. Let us input the following initial coordinate and parameter values:

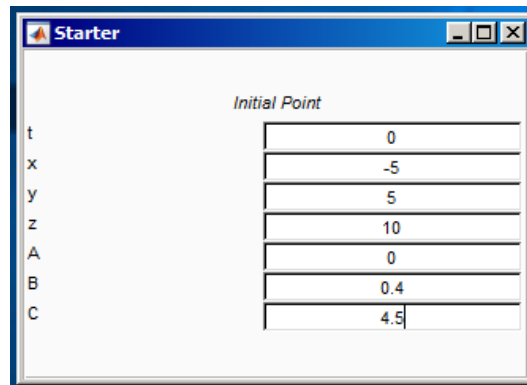


Figure 3: Starter window for integration.

```
x      -5.0
y      5.0
z     10.0

A      0.0
B      0.4
C      4.5
```

(see Figure 3).

We will use the default numerical parameters of the **Integrator**, except of the **Interval**. Input in the **Integrator** window:

```
Interval    100
```

to get Figure 4. Check that the default method of integration is **ode45**. You can inspect available methods by clicking the **Method** menu button.

5 3D visualization

To visualize the orbits, we have to open at least one more window and setup plotting attributes.

5.1 3D graphic window

In the main MATCONT window, select **Window/Output|Graphic|3D plot**. The first **Plot3D** window appears.

Plot3D windows are used to represent solution curves by their projections to 3-dimensional spaces endowed with right-handed rectangular Cartesian coordinate systems. 3D space is specified by three variables whose values are plotted along axes. The visible part of the space is given by minimum and maximum values for each axis. By default, the visibility limits are from 0 to 1 for all axes.

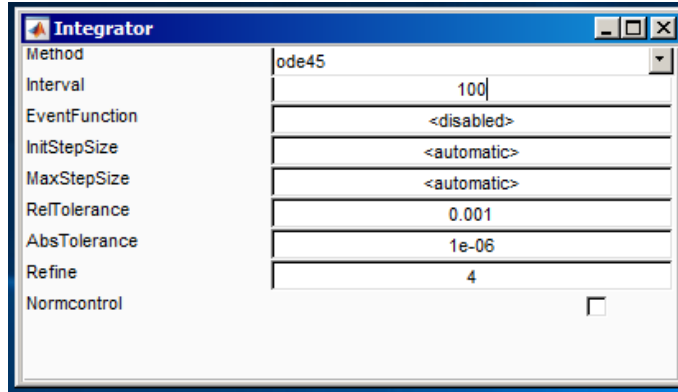


Figure 4: Integrator window.

5.2 Variables on axes

First, we need to choose the variables along the axes and the ranges of displayed values along these axes. This can be done by selecting the coordinates x , y , and z and the ranges

$$-9 \leq x \leq 9, \quad -9 \leq y \leq 9, \quad -2 \leq z \leq 11.$$

in the **Layout** window (see Figure 5). This window is accessible via the menu **MatCont|Layout** in the **Plot3D** window. Click **OK** to confirm the choice of variables and ranges.

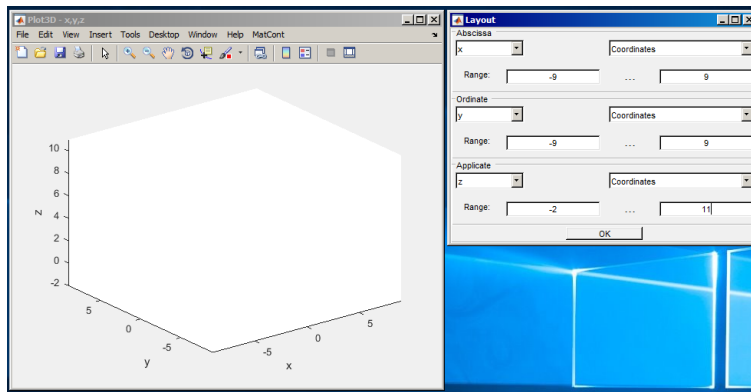


Figure 5: Selecting variables and ranges along the axes.

The graphics window **Plot3D** is updated automatically with the input and the new names of the axes become visible.

6 Integrating orbits

Now we are ready to start numerical integration. We will perform multiple simulations of the Rössler system at different values of a parameter. We will see that the behaviour of the orbits changes qualitatively, suggesting that bifurcations happen in between these parameter values.

6.1 Start computation

Input the **Compute|Forward** menu command in the main window to start integration. A **Control** window is opened automatically. The computation can be paused or stopped by clicking the corresponding button in that window. If paused, the computation can be resumed by clicking the corresponding button as shown in Figure 6.

Also, you can interrupt the computation at any time by pressing the **Esc** key on the keyboard.

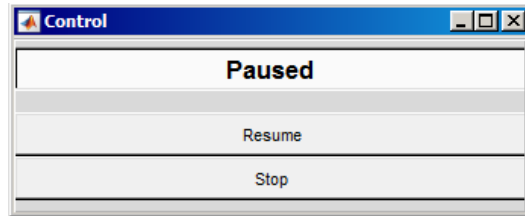


Figure 6: Control window.

The integration will start and shortly after produces Figure 7. The orbit tends to a stable equilibrium.

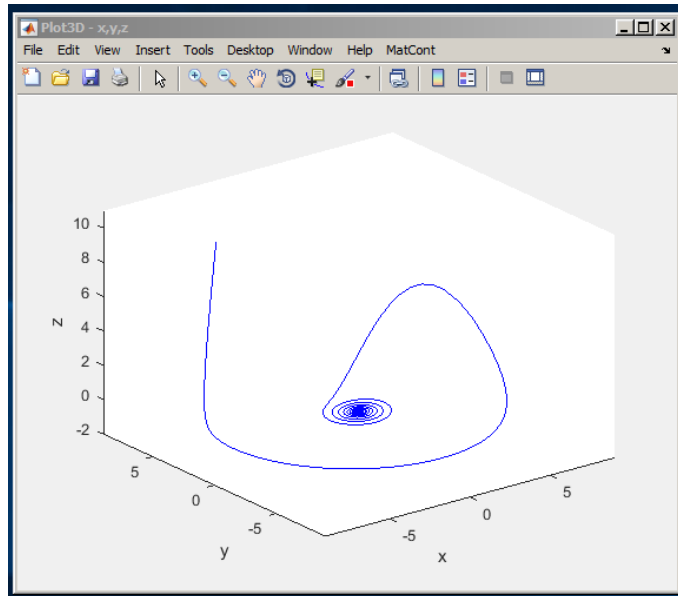


Figure 7: An orbit in the **Plot3D** window converging to a stable equilibrium.

When the computation is finished, the **Control** window looks like Figure 8. You can now either close the **Control** window or press the **View Result** button to inspect the output in a **Data Browser** window, see Figure 9. By pressing the **View CurveData** button in the **Data Browser** window one obtains a spreadsheet view of all points computed during the time integration, see Figure 10.

6.2 Other parameter and coordinate values

Clear the **Plot3D** window by selecting **MatCont|Clear** menu option there. Change the parameter **A** value to 0.2 – while keeping all other parameters unchanged – and the initial value of **z**

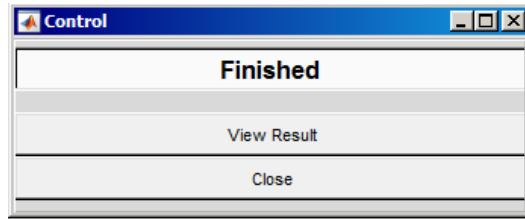


Figure 8: The **Control** window when the computation is finished.

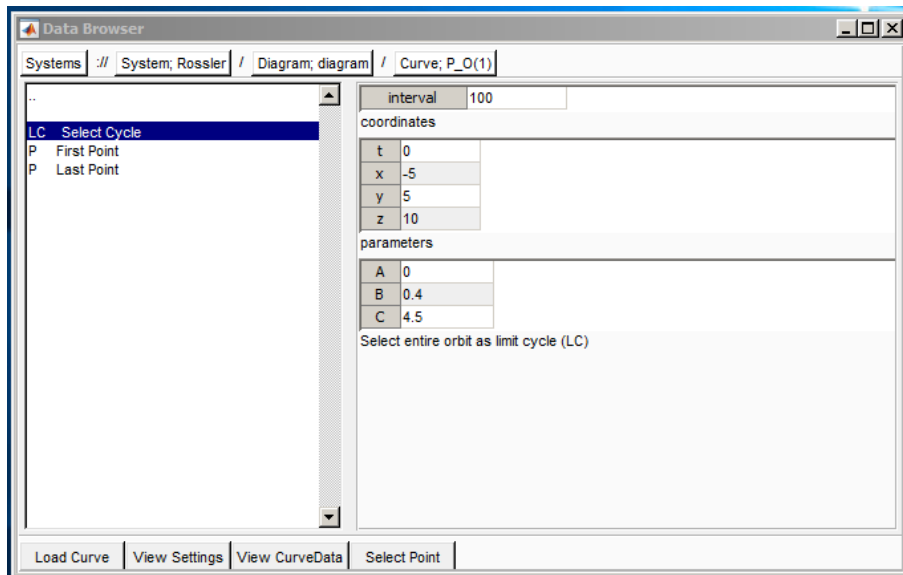


Figure 9: Inspect the output of the computation in a **Data Browser** window.

	1	2	3	4	5	6	7
t	0	0.0052	0.0104	0.0155	0.0207	0.0441	0.0674
x	-5	-5.0763	-5.1500	-5.2212	-5.2899	-5.5726	-5.8154
y	5	4.9739	4.9474	4.9206	4.8933	4.7664	4.6333
z	10	9.5078	9.0357	8.5831	8.1496	6.4135	5.0037

Figure 10: A spreadsheet view of the points computed by time integration.

to 1. Start the new integration by **Compute|Forward**. The computed orbit now tends to a stable periodic orbit (*limit cycle*), see Figure 11. This stability loss by the equilibrium with the generation of a limit cycle is called the (supercritical) *Andronov-Hopf bifurcation*.

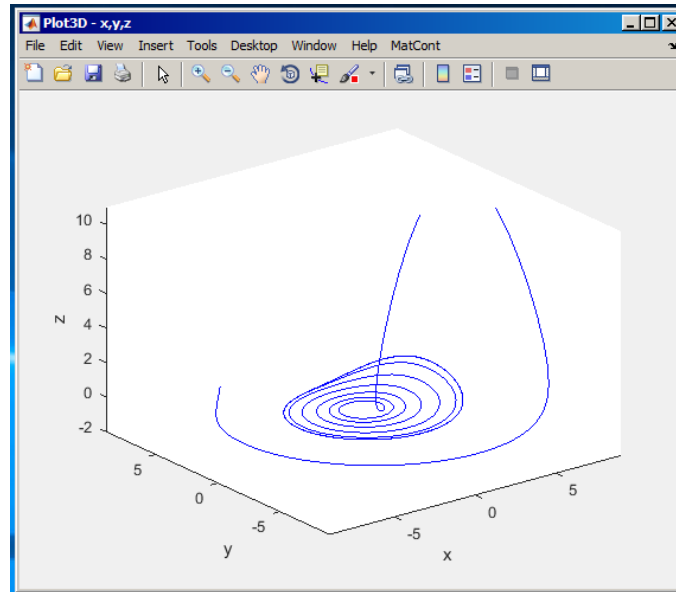


Figure 11: An orbit in the **Plot3D** window converging to a stable limit cycle at $A = 0.20$.

Repeat the integration at A equal 0.25. You should get Figure 12 showing another orbit tending to a limit cycle but slowly.

When we now repeat the integration for A equal to 0.3, we notice that the orbit tends to a limit cycle making two turns before closure. The appearance of a stable cycle with approximately doubled period, while the primary cycle becomes unstable, is called the (supercritical) *period-doubling bifurcation*. To erase transient behaviour, select **MatCont|Clear** in the **Plot3D** window and then continue the integration via the menu selection **Compute|Extend** in the main window. You should obtain Figure 13 (left panel after the initial integration and right panel without showing the transient).

Increasing the parameter A to 0.315, results in a more complicated cycle making four global turns before closure, see and reproduce Figure 14. We can conclude that the doubled cycle undergoes the next period-doubling bifurcation from a cascade of such bifurcations.

Finally, change A into 0.36, set the **Interval** in the **Integrator** window to 500, clear the graphic window, and start computation. After some time, you should see a *chaotic attractor* as in Figure 15.

7 Plot manipulation

7.1 Another viewpoint

You can change the viewpoint in the **Plot3D** window using the standard MATLAB facilities. If you press the **Rotate 3D** button in the Tools section of the menu bar of the **Plot3D** window, then the coordinate system can be rotated with the help of the mouse. A rotated orbit is shown in Figure 16.

To view the orbit in the original projection, use the standard MATLAB tools.

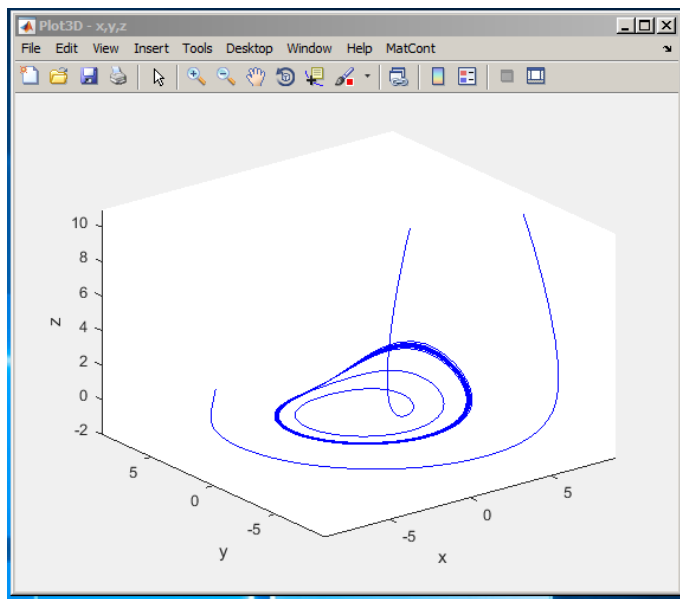


Figure 12: An orbit in the **Plot3D** window converging to a stable limit cycle at $A = 0.25$.

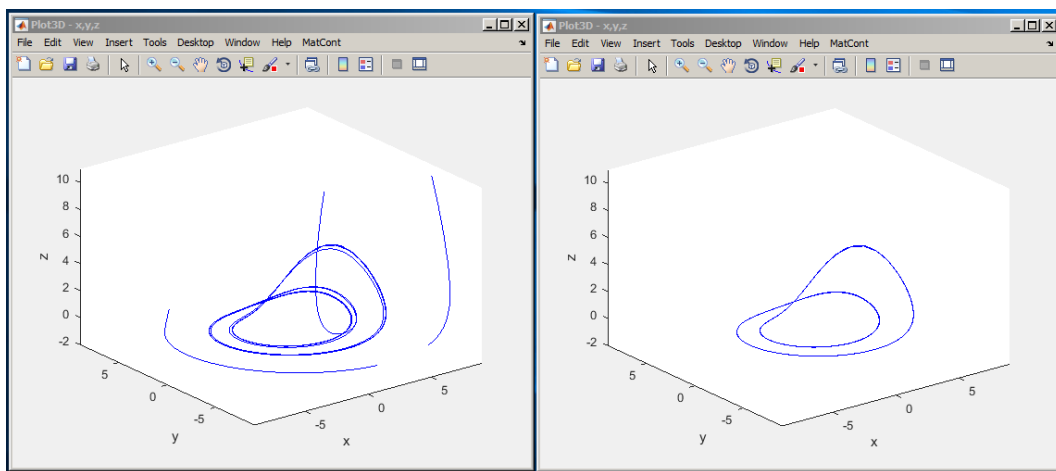


Figure 13: An orbit in the **Plot3D** window converging to a stable 2-cycle at $A = 0.3$.

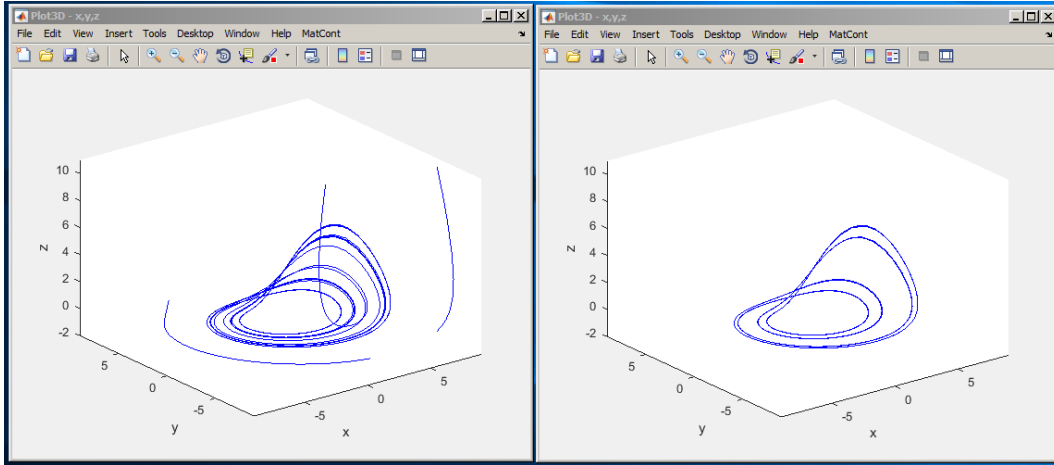


Figure 14: An orbit in the **Plot3D** window converging to a stable 4-cycle at $A = 0.315$.

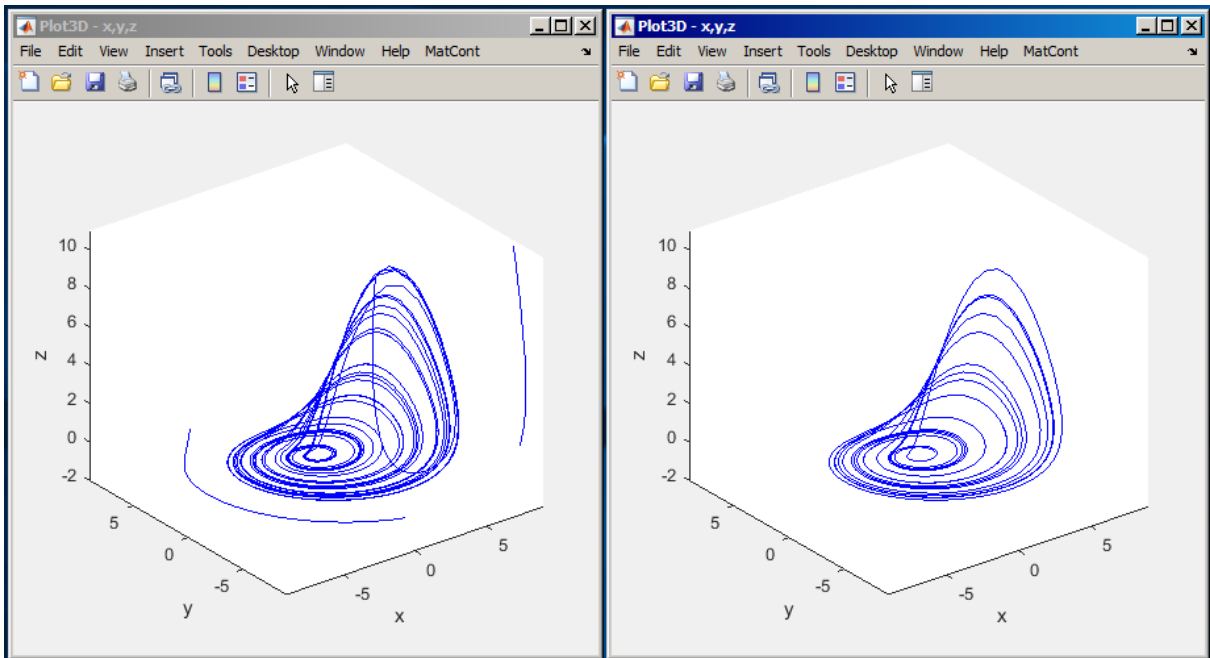


Figure 15: A chaotic attractor in the **Plot3D** window at $A = 0.36$.

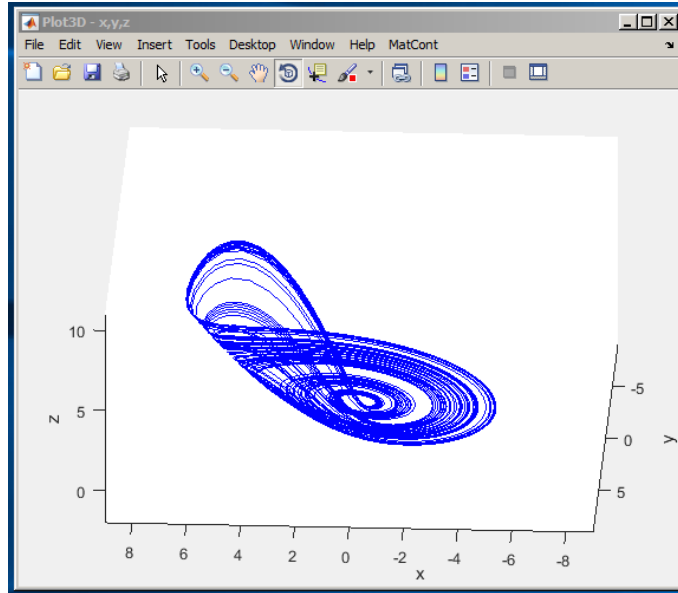


Figure 16: Another viewpoint on the chaotic attractor at $A = 0.36$.

7.2 Redraw

The computed orbit (including the transient) can be redrawn without recomputation using commands from the **Plot3D** window, namely: **Plot|Clear** followed by **Plot|Redraw Curve**.

7.3 Export a figure

You can save the produced figure in various formats using the standard MATLAB tools. For example, to produce an Encapsulated PostScript file corresponding to Figure 16, click **File|Save As..** button in the **Plot3D** window and select **EPS** type. Then specify the file name and location in the appearing dialog box, and press **Save**.

8 2D visualization

Let us plot the time series corresponding to the computed chaotic orbit for $0 \leq t \leq 250$. For this, open another graphic window via the **Window/Output|Graphic|2D plot** in the main **MATCONT** window.

A new window **Plot2D** appears that has a structure similar to the **Plot3D** window.

Using **MatCont|Layout**, set time t as the **Abscissa** and coordinate x as the **Ordinate** and input new visibility limits along the axes:

```
t      0  250
x     -8   8
```

The layout window is presented at the right of Figure 17.

Close the **Plot3D** window. In the **Plot2D** window, you can get with **MatCont|Redraw Curve** a time series shown in Figure 18. You may need to resize the window for a better view.

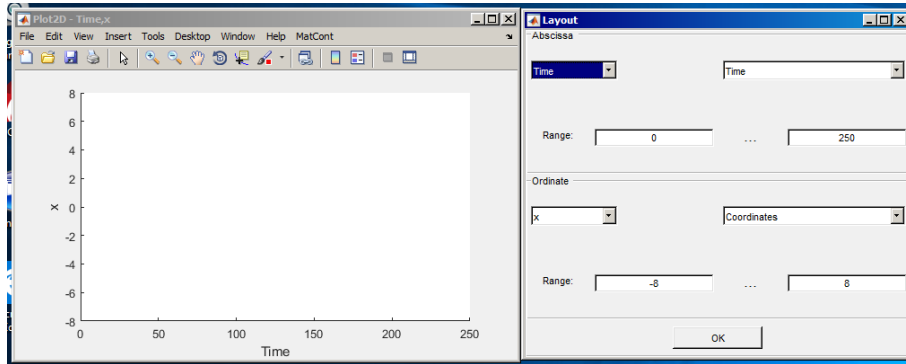


Figure 17: Coordinates and range limits in a 2D plot.

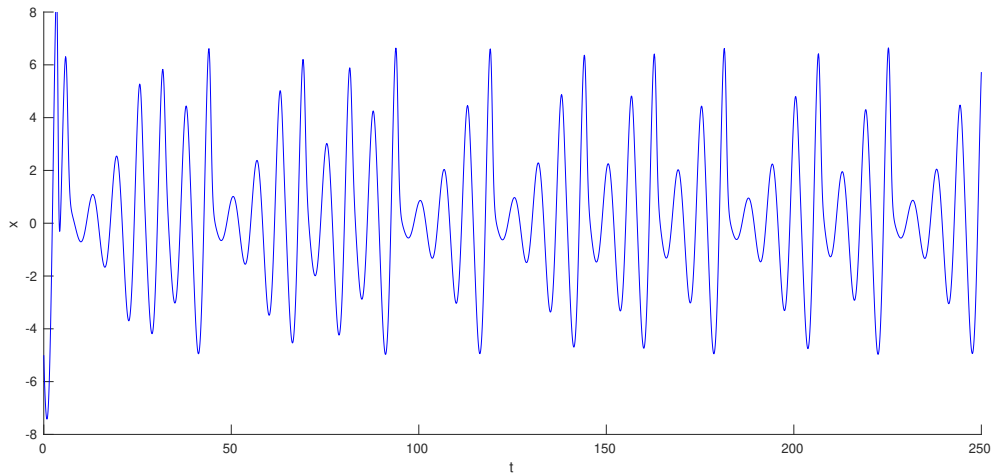


Figure 18: The first solution component versus time in the **Plot2D** window.

9 Another method of integration

The method of integration can be changed. For example, one can select `ode23s` as **Method** in the **Integrator** window. After selection, the **Integrator** window will be updated.

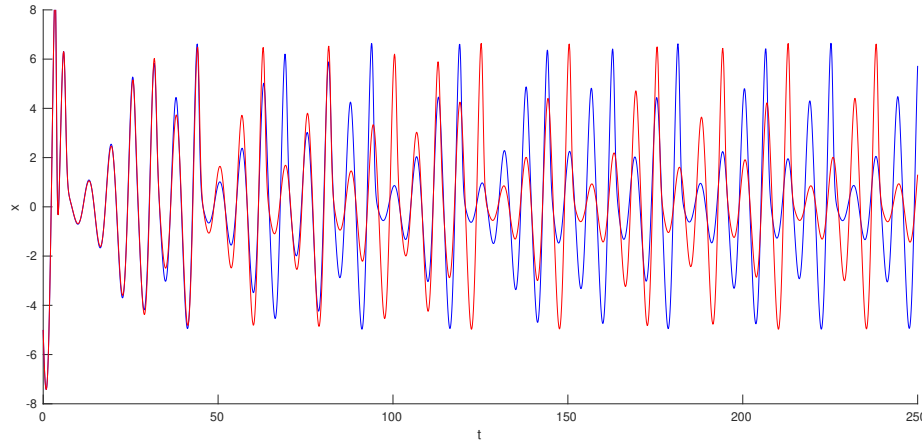


Figure 19: Two numerical solutions obtained with different integration methods. The second curve was computed with `ode23s`. The color of the second curve has been set to red to accentuate the difference.

Repeat computations with **Compute|Forward**. Notice in Figure 19 a change in the time series for big values of t due to a difference in the accuracy of the methods. This shows that one has to be careful with numerical solutions of ODEs.

Note: Normally you will get Figure 19 in one color. We used a property editor to change the color of the second curve.

10 Archive of computed solutions

By default, MATCONT keeps only a certain number of computed curves (e.g. orbits) under the standard names corresponding to their **Point type** and **Curve type**. This number is determined by the Archive Filter which can be seen and adapted by the user by pressing the **Options|Archive Filter** button, see Figure 20. With this setting, only two curves of each type are preserved.

You can manipulate the curves via **Select|Curve** menu in the main MATCONT window. This opens a **Data Browser** window as displayed in Figure 21.

It is now possible to load a curve (for example to redraw it separately), view, rename or delete it.

As an exercise, inspect a spreadsheet view of the computed points of the highlighted curve by pressing the **View** button in the **Data Browser** window.

If a curve is renamed, it will be stored permanently. As an exercise rename the two remaining orbits into `orbit1` and `orbit2`, respectively. By pressing the **New** button one returns to an empty curve in the diagram that is open in the **Data Browser** window.

Close the **Data Browser** window.

All currently available curves form a *diagram* that can be redrawn in a graphic window via **MatCont|Redraw Diagram**. If you do this in the **Plot2D** window, you should get Figure 19 again.

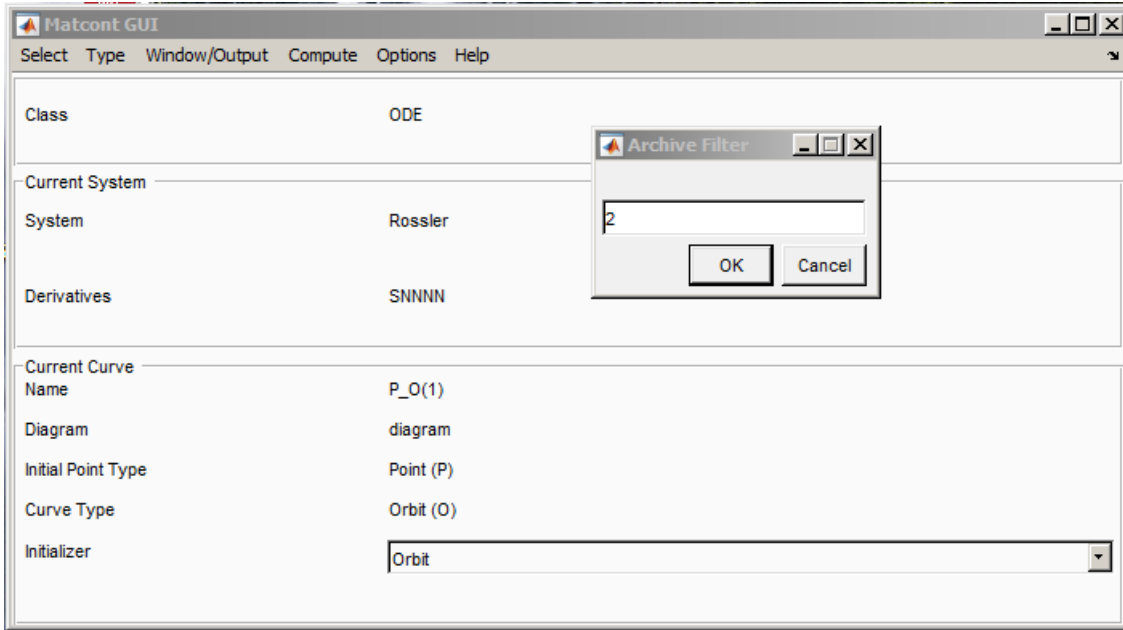


Figure 20: The Archive Filter is set to 2.

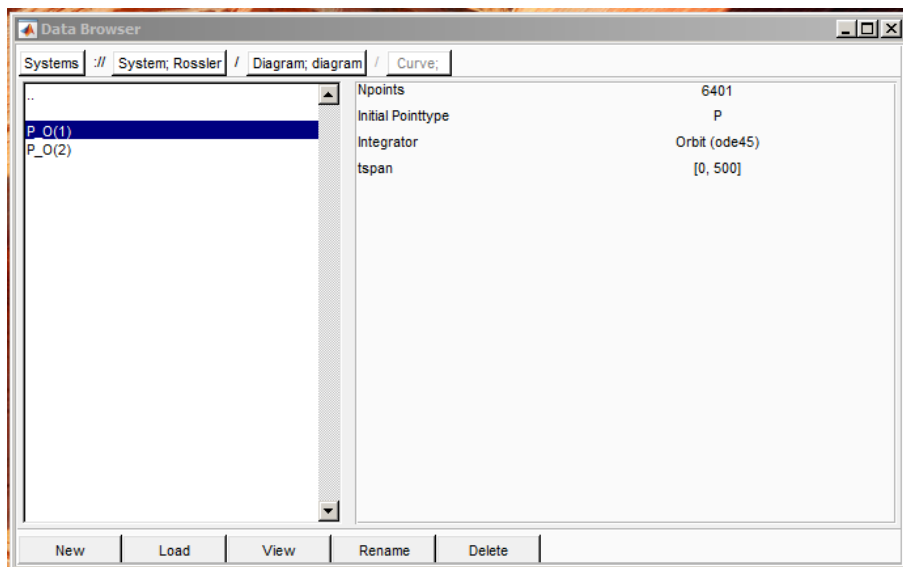


Figure 21: The Data Browser window opened by pressing the Systems|Curve button.

11 Additional Problems

A. Consider the *logistic equation*

$$\dot{x} = x \left(1 - \frac{x}{\alpha} \right)$$

with positive α , e.g. $\alpha = 2$. Plot solutions $(t, x(t))$ with different initial conditions $x(0)$.

B. *Van der Pol's equation* is given by

$$\ddot{x} - \alpha(1 - x^2)\dot{x} + x = 0.$$

Rewrite the equation as a system of two first-order ODEs by introducing $y = -\dot{x}$. Plot its orbits for $\alpha = 0$ and $\alpha = 0.5$ in the phase space (x, y) .

C. Simulate the *Lorenz chaotic system*

$$\begin{cases} \dot{x} &= \sigma(y - x), \\ \dot{y} &= rx - y - xz, \\ \dot{z} &= -bz + xy, \end{cases}$$

for $\sigma = 10$, $b = \frac{8}{3}$, $r = 28$. Make 3D phase plots as well as plot time-series of z for various initial conditions.

D. Consider the following *Langford system*

$$\begin{cases} \dot{x} &= (\lambda - b)x - cy + xz + dx(1 - z^2), \\ \dot{y} &= cx + (\lambda - b)y + yz + dy(1 - z^2), \\ \dot{z} &= \lambda z - (x^2 + y^2 + z^2), \end{cases}$$

where $b = 3$, $c = \frac{1}{4}$ and $d = \frac{1}{5}$.

1. Integrate this system in MATCONT for $\lambda = 1.5, 1.9, 2.01$. Always start from $x_0 = y_0 = 0.1, z_0 = 1$.
2. Briefly describe what you see in each case.
3. Could you support your conclusions by some other numerical and/or analytical arguments? *Hint*: Introduce cylindrical coordinates (r, φ, z) so that $x = r \cos \varphi$, $y = r \sin \varphi$.