

TUTORIAL I:
Using MATCONT for numerical integration of ODEs

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

September 1, 2009

This session 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.

We will study *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 directory `matcont` of your system. Start MATLAB and change the current directory to `matcont`. Start MATCONT by typing

`matcont`

in MATLAB command line window and press Enter¹.

¹If you receive complains about a compiler, enter `mex -setup` and choose the `lcc`, the MATLAB C-compiler. Then start MATCONT again.

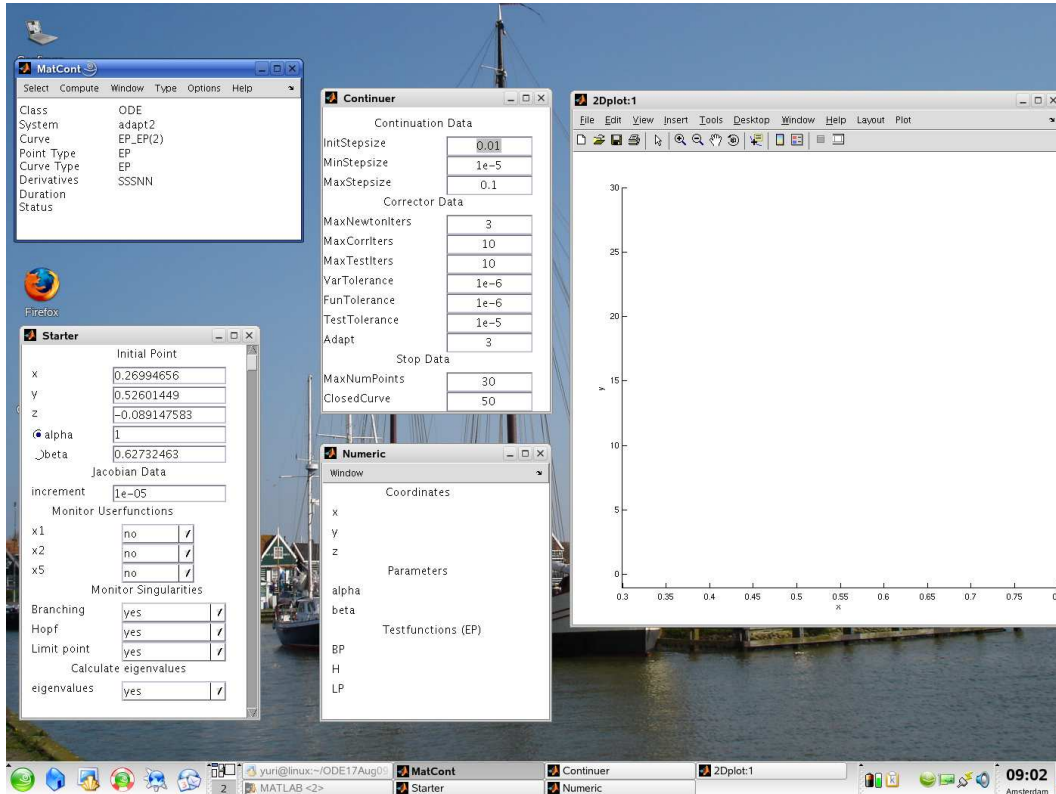


Figure 1: A typical matcont startup screen.

You will get several windows related to the default ODE system “adapt2”, like in Figure 1. The main window is called **MatCont** and has several menus. For instance, to end your MATCONT session, choose **Exit** item in **Select** menu. Hereafter this operation will be indicated with **Select|Exit**. This time, press **No** button.

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**: AA, BB, CC , and the name for **Time**: t (default).

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

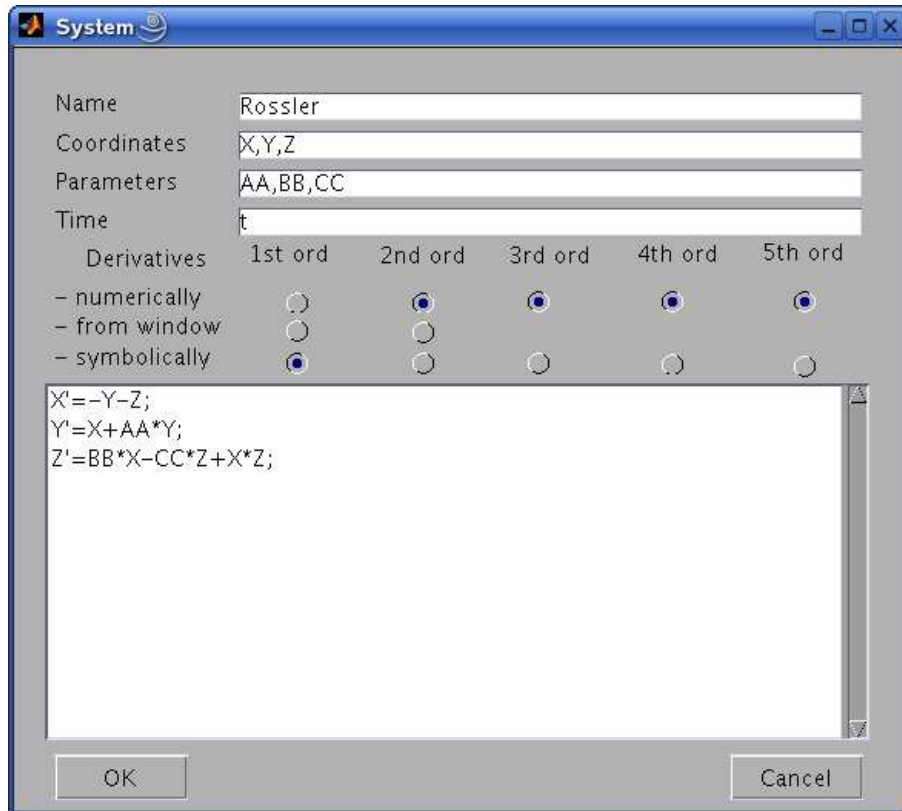


Figure 2: Specifying a new model.

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

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.

```

X'=-Y-Z
Y'=X+AA*Y
Z'=BB*X-CC*Z+X*Z

```

Avoid typical mistakes:

- Some names (e.g. C) are reserved in MATLAB and cannot be used.
- Do not put spaces between coordinate and parameter names.
- 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 that **Rosslor** becomes the current **System** of MATCONT. If selected, the information field **Derivatives** shows the string **SNNNN** 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**, select this system in the appearing **Systems** window, and press **Edit** button. The inputed 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, input **Type|Curve|Orbit**³. 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:

X	-5.0
Y	5.0
Z	10.0
AA	0.0
BB	0.4
CC	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.

³Actually, it is default.

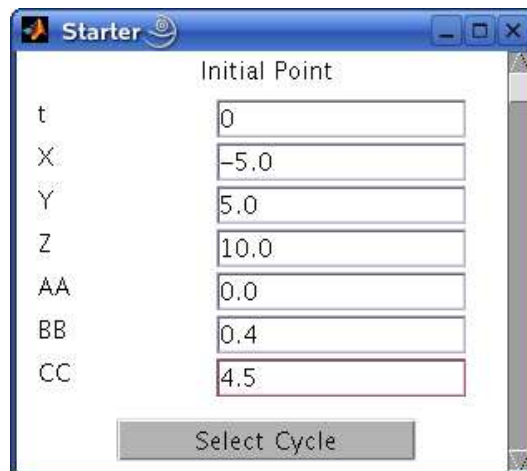


Figure 3: Starter window for integration.

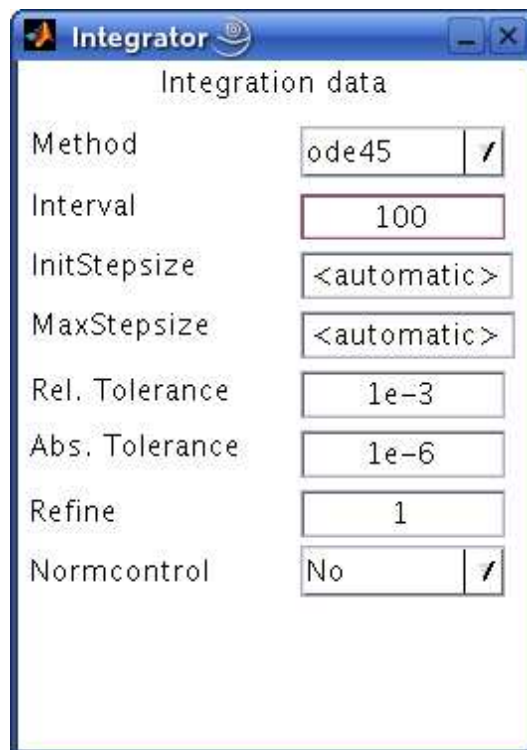


Figure 4: Integrator window.

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|Graphic|3Dplot** . The first **3Dplot** window appears.

3Dplot 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. 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.

5.2 Variables on axes

First, we need to choose the variables along the axes. This can be done by selecting the coordinates **X**, **Y**, and **Z** in the **Variables on axes** window using the pull-down menus (see Figure 5). Click

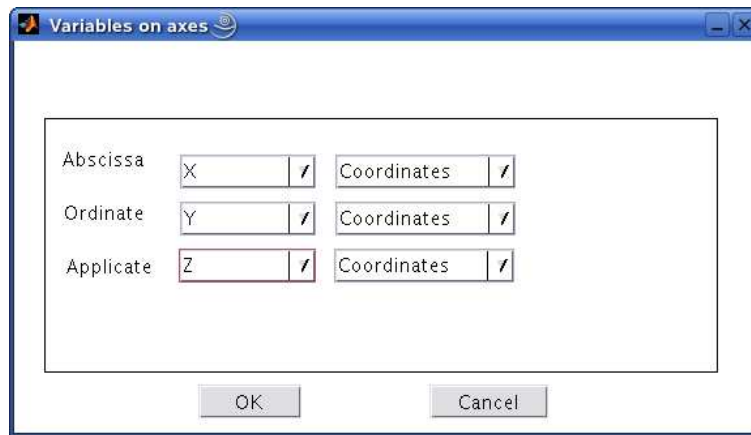


Figure 5: Selecting variables along the axes.

OK to confirm the variable selection. The new names of the axes will become visible.

This window is also accessible via the menu **Layout|Variables on axes** in the **3Dplot** window.

5.3 Plotting region

To define the visibility limits along the axes, open the **3D Plotting region** window by selecting **Layout|Plotting region** in the same **3Dplot** window. Let us consider the region

$$-8 \leq x \leq 8, \quad -8 \leq y \leq 8, \quad -1 \leq z \leq 10.$$

Thus type the visibility limits as in Figure 6 and close the window.

The graphics window **3Dplot** is updated automatically with the input.

6 Integrating orbits

Now we are ready to start numerical integration.

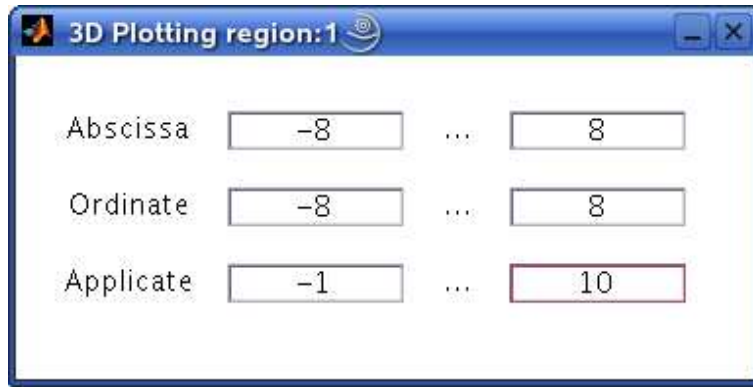


Figure 6: Setting visibility limits.

6.1 Start computation

Input the **Compute|Forward** menu command in the main window to start integration. The computation can be paused/resumed/stopped by clicking the corresponding button in the appearing small control window shown in Figure 7. Also, you can interrupt the computation at any time by pressing **Esc** key on the keyboard.



Figure 7: Control window.

The integration will start and shortly after produces Figure 8. The orbit tends to a stable equilibrium.

6.2 Other parameter values

Clear the **3Dplot** window by selecting **Plot|Clear** menu option there. Change the parameter **AA** value to 0.2 – while keeping all other parameters unchanged – and start the new integration by **Compute | Forward**. The computed orbit now tends to a stable periodic orbit (*limit cycle*), see Figure 9. This stability loss by the equilibrium with the generation of a limit cycle is called the (supercritical) *Andronov-Hopf bifurcation*.

Repeat the integration at **AA** equal 0.25. You should get Figure 10 showing another orbit tending to a limit cycle but slowly.

When we now repeat the integration for **AA** 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 **Plot|Clear** in the **3Dplot** window and then continue the integration via the menu selection **Compute|Extend** in the main window. You should obtain Figure 11 (case (a) after the initial integration and (b) without showing the transient).

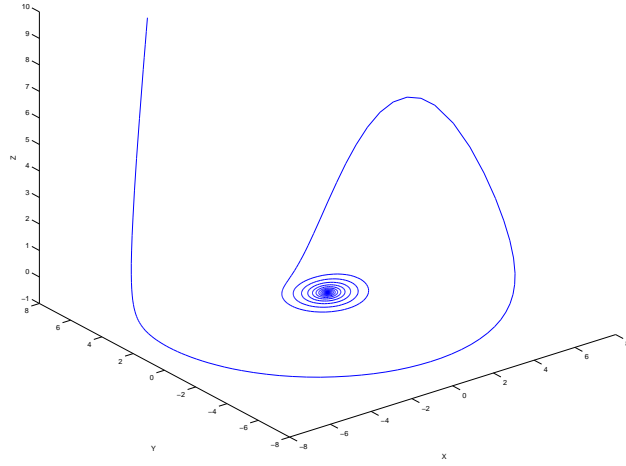


Figure 8: An orbit in the **3Dplot** window converging to a stable equilibrium.

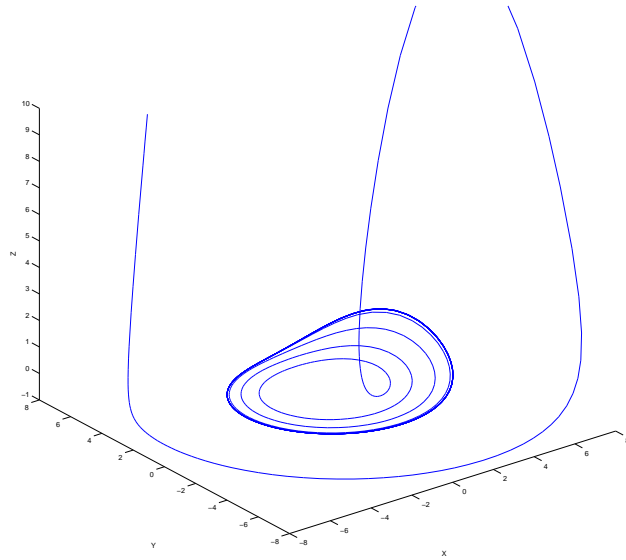


Figure 9: An orbit in the **3Dplot** window converging to a stable limit cycle at $A = 0.20$.

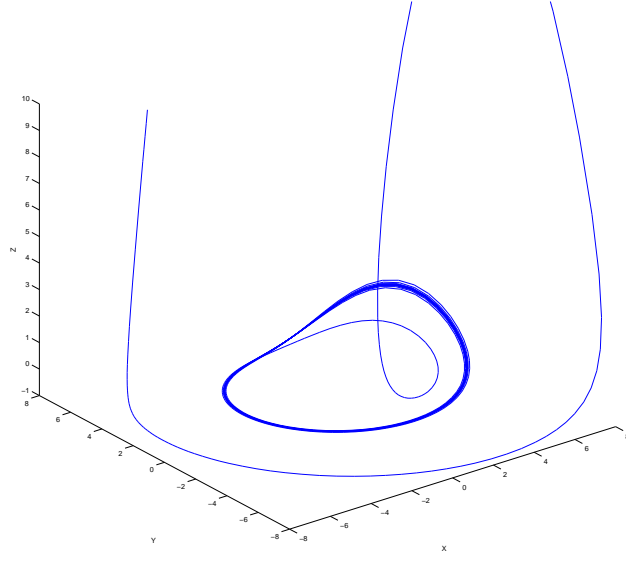


Figure 10: An orbit in the **3Dplot** window converging to a stable limit cycle at $A = 0.25$.

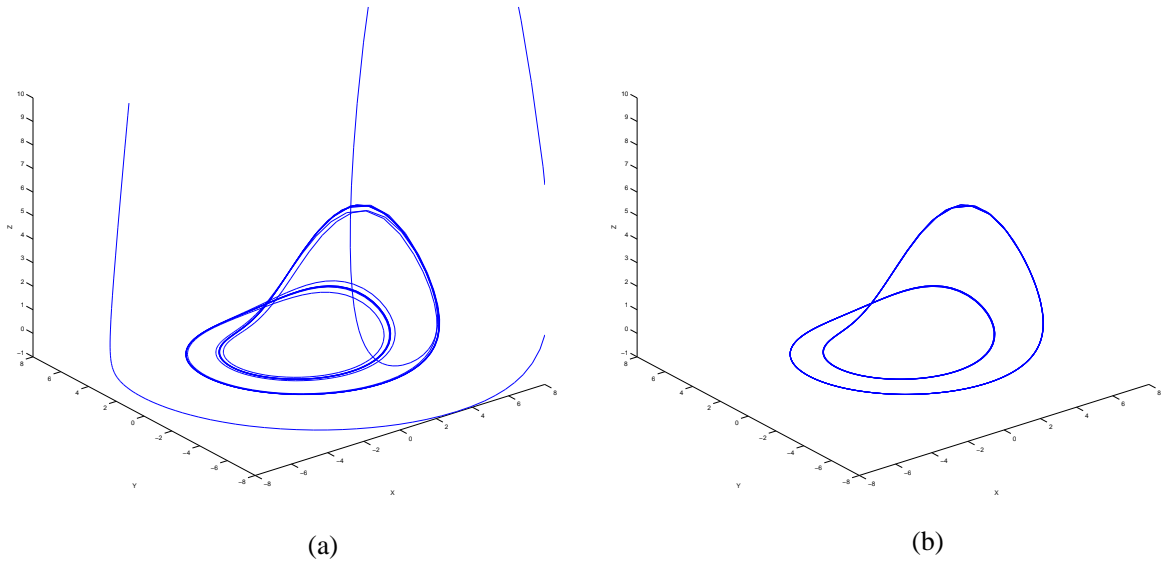


Figure 11: An orbit in the **3Dplot** window converging to a stable 2-cycle at $A = 0.3$.

Increasing the parameter **AA** to 0.315, results in a more complicated cycle making four global turns before closure, see and reproduce Figure 12. We can conclude that the doubled cycle undergoes the next period-doubling bifurcation from a cascade of such bifurcations.

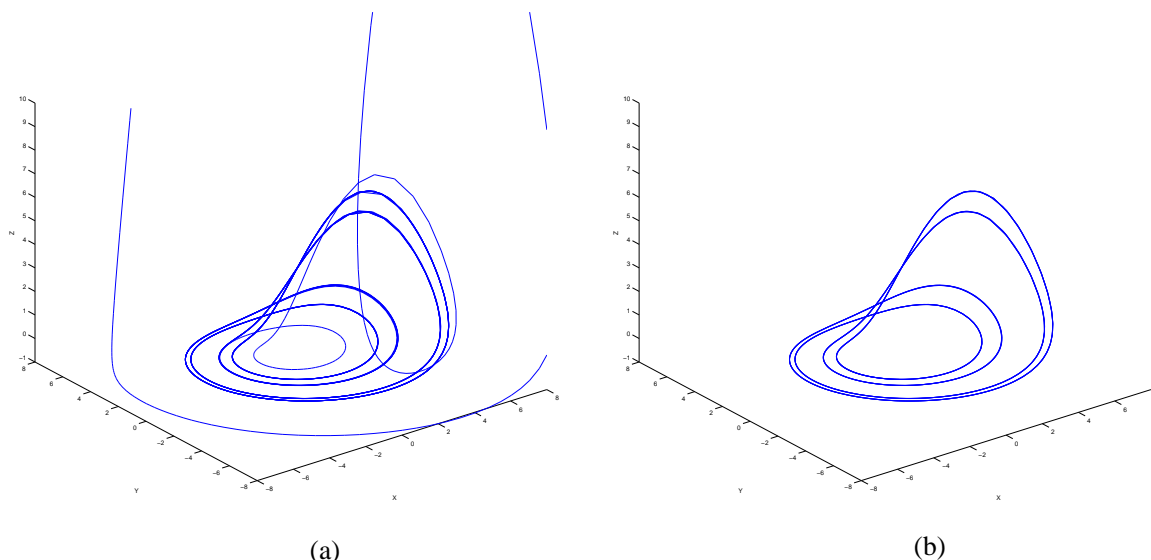


Figure 12: An orbit in the **3Dplot** window converging to a stable 4-cycle at $A = 0.315$.

Finally, change **AA** 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 13.

7 Plot manipulation

7.1 Another viewpoint

You can change the viewpoint in the **3Dplot** window using the standard MATLAB facilities. If you press the **Rotate 3D** button in the menu bar of the **3Dplot** window, then the coordinate system can be rotated with the help of the mouse. A rotated orbit is shown in Figure 14.

To view the orbit in the original projection, click the right mouse button and select **Reset to Original View**.

7.2 Redraw

The computed orbit (including the transient) can be redrawn without recomputation using commands from the **3Dplot** 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 14, click **File|Save As..** button in the **3Dplot** window, select **EPS** type, the file name and location in the appearing dialog box, and then press **Save**.

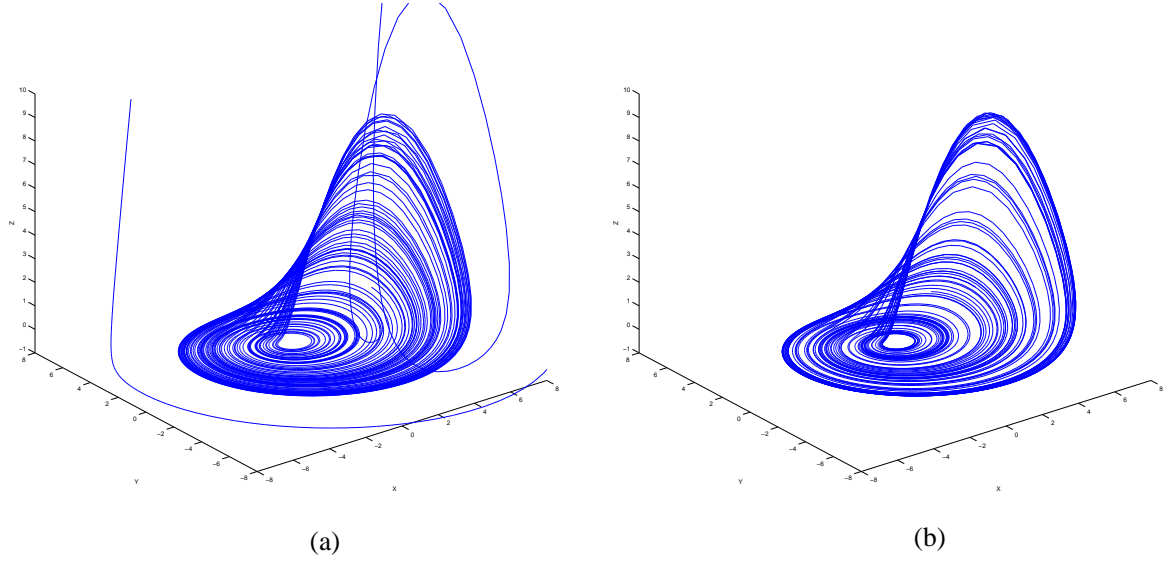


Figure 13: A chaotic attractor in the **3Dplot** window at $A = 0.36$.

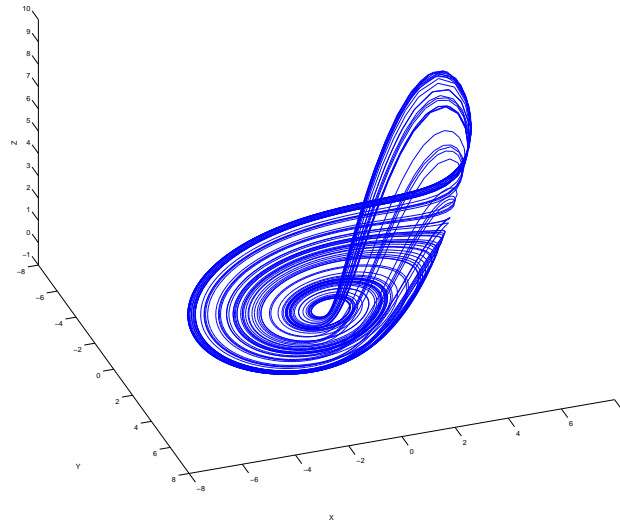


Figure 14: Another viewpoint on the chaotic attractor at $A = 0.36$.

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|Graphic|2Dplot** in the main **MatCont** window.

A new window **2Dplot** appears that has a structure similar to the **3Dplot** window. Set time **t** as the **Abscissa** and coordinate **X** as the **Ordinate** in the **Variables on axes** window and click **OK** button.

Using **Layout|Plotting region**, input new visibility limits along the axes:

t	0	250
X	-8	8

and close the **3D Plotting region** window. In the **2Dplot** window you should immediately get a time series shown in Figure 15. You may need to resize the window for a better view.

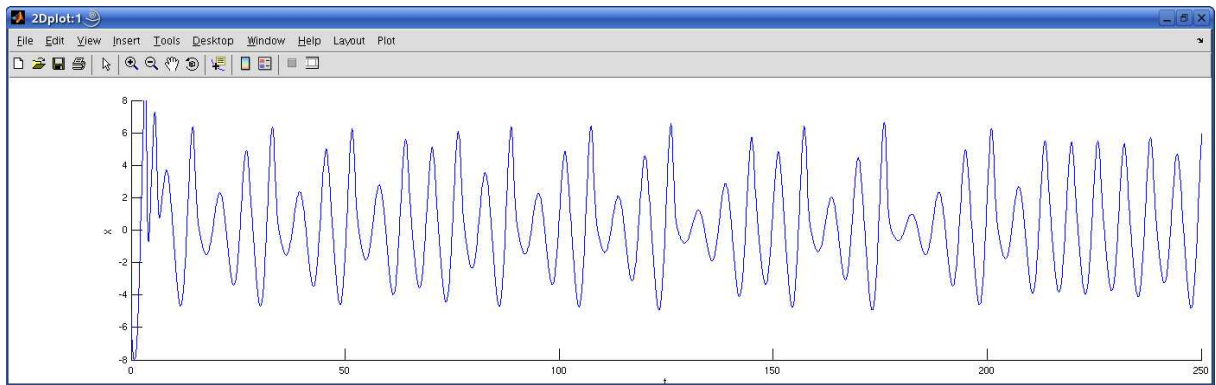


Figure 15: The first solution component versus time in the **2Dplot** window.

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.

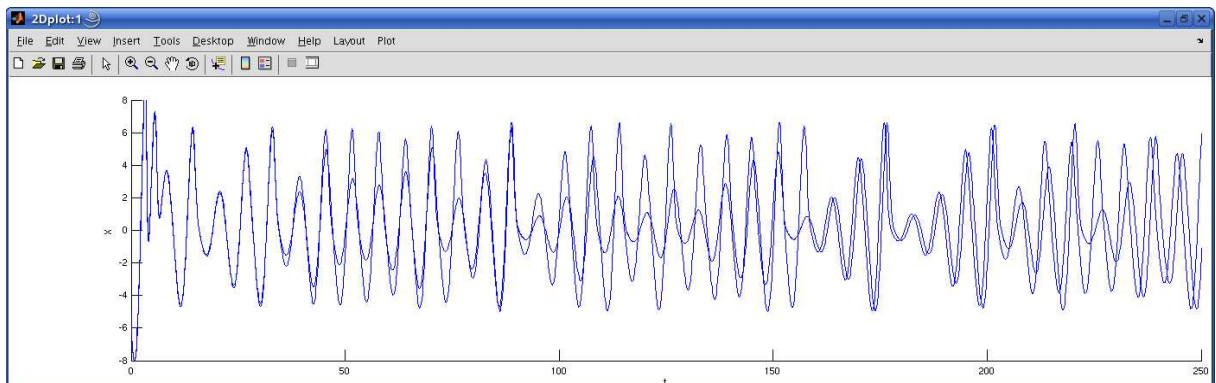


Figure 16: Two numerical solutions obtained with different integration methods.

Repeat computations with **Compute|Forward**. Notice in Figure 16 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.

10 Archive of computed solutions

By default, MATCONT keeps three last computed curves (e.g. orbits) under the standard names corresponding to their **Point type** and **Curve type**. You can manipulate the curves via **Select|Curve** menu in the main **MatCont** window.

Curves window will appear, where the stored curves are listed, in our case

P_0(1)

P_0(2)

P_0(3)

Using the **Select** button and the **Actions** pull-down menu of the **Curves** window, it is possible to select a curve (for example to redraw it separately), rename or delete it. If a curve is renamed, it will not be overwritten and will be stored permanently.

As an exercise, find out which curve corresponds to the transient towards a periodic orbit and delete it. Then rename two remaining orbits into **orbit1** and **orbit2**, respectively (see Figure 17). Leave the curve archive window.



Figure 17: Curve archive.

All currently available curves form a *diagram* that can be redrawn in a graphic window via **Plot|Redraw diagram**. If you do this in the **2Dplot** window, you should get Figure 16 again.

11 Additional Problems

A. Consider the *logistic equation*

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

with positive α . Plot solutions $(t, x(t))$ with different initial conditions $x(0)$.

B. *Van der Pol's equation* is

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

Rewrite the equation as a system of two first-order ODEs and plot its orbits for $\alpha = 0$ and $\alpha = 0.5$.

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 $\sigma = 10$, $b = \frac{8}{3}$, $r = 28$.

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 $\lambda = 1.5, 1.9, 2.01$. Start always from $x_0 = y_0 = 0.1, z_0 = 1$.
2. Briefly describe what do you see in each case.
3. Could you support your conclusions by some other numerical and/or analytical arguments ?