

# A MatCont Tutorial

## Learning the GUI by exploring a Neural Mass Model

Hil Meijer

November 2023\*

"If you want to get credit for solving a complicated mathematical problem, you will have to provide a complete proof. But if you're trying to make something as easy as possible, you want to make it foolproof—so simple even a fool couldn't screw it up." <sup>1</sup>

### 1 Introduction

This tutorial explains the basics for using the numerical bifurcation package MatCont by going through an example. The focus is on using the software, so we assume a basic knowledge of bifurcation theory. As with many things one learns, you will only master it by performing the procedures yourself. So, hands-on!

We cover the following procedures in this tutorial:

- Download and Installation
- System definition
- Simulations
- Continuation of equilibria in one parameter
- Continuation of codimension one bifurcations of equilibria in two parameters; Limit Point and Hopf
- Starting Continuation of a Limit Cycle from a Hopf point
- Starting Continuation of a Limit Cycle from a time simulation

This tutorial is a condensed overview, see the tutorial sessions 1-5 at <https://sourceforge.net/projects/MatCont/files/Documentation/MatContODE/BasicTutorials/>, or the manual at Sourceforge for more information. If at some point you want to cite our work, then refer to *New features of the software MatCont for bifurcation analysis of dynamical systems*. A. Dhooge, and W. Govaerts, Yu.A. Kuznetsov, H.G.E. Meijer and B. Sautois, *Mathematical and Computer Modelling of Dynamical Systems*, Vol. 14, No. 2, pp 147-175, 2008.

Writing this tutorial, a few "harmless" bugs were encountered, you can ignore these.

- System definition complains about C1,C3 for 2nd and 3rd order derivatives, yet the resulting code is correct.
- Normal form coefficients for Period-Doubling appear to be zero, which should not be the case. Not treated in this tutorial, so we ignore this.
- Limit Point continuation shows some matrix is singular.

\*Version 1: written for version 4p2, February 2014, Exeter, updated to 7p4

<sup>1</sup>Every one computer is different from the other. So, although these instructions have been checked step by step, do not be surprised if your computations take a different direction. Just try again, but now in the other direction.

## 2 Getting started/Installation

1. Please let Matlab be installed, and download the latest version of MatCont from this link: <https://sourceforge.net/projects/MatCont/files/MatCont/MatCont7p4/MatCont7p4.zip>. Unzip this package to a folder of your choice. This package contains the GUI version, but command-line use is also possible. To show this, we have written a Matlab script that executes all steps from the command line.
2. We use compiled mex-files to speed up the continuation of limit cycles and their bifurcations. For 64-bit Windows, the necessary files are included in the package. For other Operating Systems:
  - 2a. For Macs or 32-bit Windows, you can download these files from: [https://sourceforge.net/projects/MatCont/files/Auxiliaries%28MEX\\_Files%29/MEXMac64/](https://sourceforge.net/projects/MatCont/files/Auxiliaries%28MEX_Files%29/MEXMac64/). Put these seven files into the folder LimitCycle of your MatCont folder.
  - 2b. If you're not sure which files you need, then type `mexext` on the Matlab command line.
  - 2c. For other platforms, you will need a compiler. On Linux: `gcc` is available, and with `mex -setup`, you can select this compiler. For some Mac versions, one needs to re-install XCode so that `gcc` is found (even if it is already there!).
3. Change your Matlab working directory to the Matcont folder and type `MatCont`. The MatCont GUI window should appear, we refer to this as the 'Main' window.

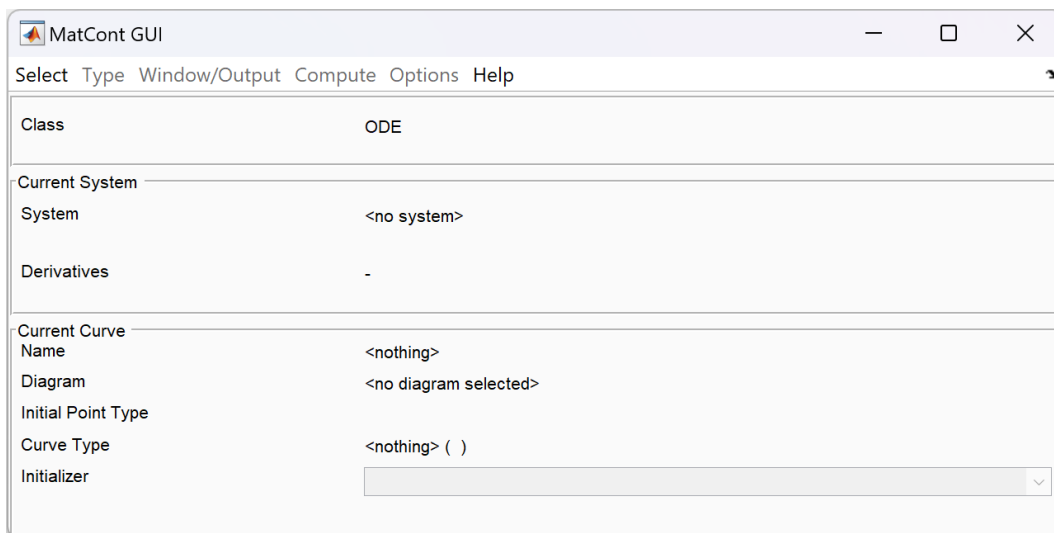


Figure 1: A typical MatCont startup screen. Depending on your earlier actions, more windows may appear.

More windows could open with a standard system called `adapt2`. A typical screenshot is shown in Figure 1. One window is titled `MatCont` and has several menu options. For instance, to end your MatCont session, choose 'Select' in the MatCont window and then 'Exit'. Hereafter, we will indicate this with *MatCont:Select|Exit*. Alternatively, one can click on the close button of the MatCont window.

### 3 Equilibria in a modified Jansen-Rit Neural Mass Model

In this tutorial, we investigate bifurcations of steady states and periodic orbits in the Jansen-Rit model with an additional slow inhibitory component [2]. The model equations are given by

$$\begin{cases} y_0'' + 2ay_0' + a^2y_0 = AaS(y_1 - \theta y_2 - (1 - \theta)y_3), \\ y_1'' + 2ay_1' + a^2y_1 = Aa(II + C_2S(C_1y_0)), \\ y_2'' + 2b_fy_2' + b_f^2y_2 = B_f b_f C_4 S(C_3y_0), \\ y_3'' + 2b_sy_3' + b_s^2y_3 = B_f b_f C_4 S(C_3y_0). \end{cases} \quad (1)$$

The sigmoidal function  $S$  is given by  $S(u) = \frac{2e_0}{1 + \exp(r(v_0 - u))}$  and the constants satisfy  $C_2 = 0.8C_1$  and  $C_4 = C_3 = 0.25C_1$ .

#### 3.1 Specifying a new model

First, we are going to specify the differential equations of the system. MatCont works on autonomous first-order systems  $x' = F(x, p)$  with  $(x, p) \in \mathbb{R}^{n \times m}$ . So we introduce dummy variables  $y_{i+4} = y_i'$  to convert (1) to an eight-dimensional system of first-order ODEs. We will create a m-file with functions we can call for simulation and continuation.

Click `(MatCont):Select Systems|New`. A window titled 'System' appears with fields 'Name system', 'Coordinates', 'Parameters', 'Time', options for derivatives and a large box. The exact appearance of the System window depends on the operating system and the availability of the Matlab Symbolic Toolbox. If the latter is present, the lowest line will read 'symbolically'. It is preferred to compute first-, second- and third-order derivatives symbolically as this is the most reliable and fastest option. The first- and second-order derivatives are heavily used in the continuation, so this speeds up the computations. The third-order derivatives are useful for computing normal forms accurately. Copy-paste all the fields from the following box to their respective fields. If that does not work, then either type it or go to <https://wwwhome.ewi.utwente.nl/~meijerhge/MT.txt> that has all fields in plain text for easy copy and paste.

### MatCont Input

```
Name      JR_slow
Coordinates y0,y4,y1,y5,y2,y6,y3,y7
Parameters II,A,a,Bs,bs,Bf,bf,e0,r,v0,theta,CC
Equations
C1=CC
C2=0.8*CC
C3=0.25*CC
C4=0.25*CC
S4=2*e0/(1+exp(r*(v0-(y1-theta*y2-(1-theta)*y3))))
S5=2*e0/(1+exp(r*(v0-(C1*y0))))
S6=2*e0/(1+exp(r*(v0-(C3*y0))))
S7=S6
y0'=y4
y4'=A*a*S4-2*a*y4-a*a*y0
y1'=y5
y5'=A*a*(II+C2*S5)-2*a*y5-a*a*y1
y2'=y6
y6'=Bf*bf*C4*S6-2*bf*y6-bf*bf*y2
y3'=y7
y7'=Bs*bs*C4*S7-2*bs*y7-bs*bs*y3
```

Select symbolic derivatives up to order 3, if possible. Otherwise, stick to Numerical derivatives. After entering all fields, it will look like Figure 2. Press **OK**. First, a small pop-up window appears stating 'Precomputing

model', indicating the progress of processing the system, and then the system definition window disappears.

**Note:** There is a warning about C1 and C3, but you can ignore this.

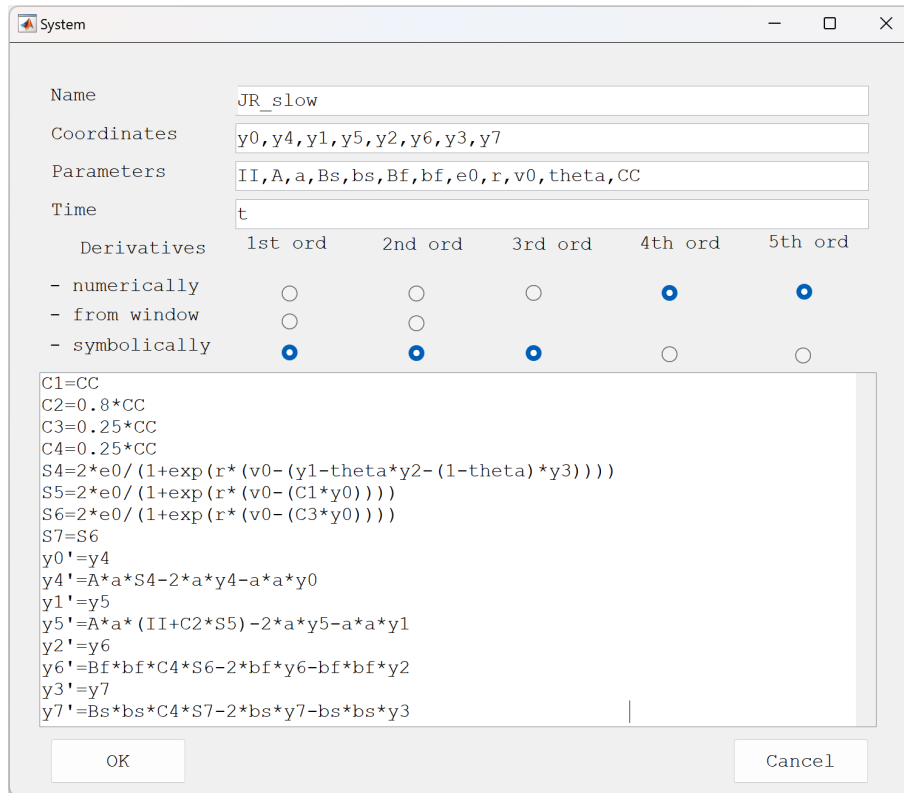


Figure 2: Specifying a new model

If you want to change the system or correct the input, choose `(MatCont):Select|Systems|Edit/Load'`, select the system you want to edit and press 'Edit'.

Common mistakes (apart from simple typos):

- Make sure that multiplication is written explicitly with `*`, or you get errors like 'Unbalanced or misused parentheses or brackets' or 'error using ==> eval, Index exceeds matrix dimensions'.
- If your model has some strange behaviour like ever-increasing variables, check whether you have specified the differential equations in the same order as the order of the coordinates.
- This example shows you can use some intermediate expressions. These should all be specified before the differential equations.

### 3.2 Time Simulations

Click `(MatCont):Type|Initial point|Point'` to initialize the computation of an orbit starting from a point. The MatCont window will now show the curve type is now P\_O. Every curve type has a similar meaning; the first label is the type of initial data given, and the second label is the type of curve computed. Two extra windows, 'Starter' and 'Integrator', appear. The Starter window specifies all initial conditions and parameter values. The Integrator window allows you to select the numerical integrator and adapt its settings. In the Starter window, set  $II = 20$ ,  $A = 3.25$ ,  $a = 100$ ,  $Bs = 8.8$ ,  $bs = 20$ ,  $Bf = 44$ ,  $bf = 100$ ,  $e0 = 2.5$ ,  $r = 0.56$ ,  $v0 = 6$ ,  $theta = 0.5$ ,  $CC = 220$ , see Figure 3.

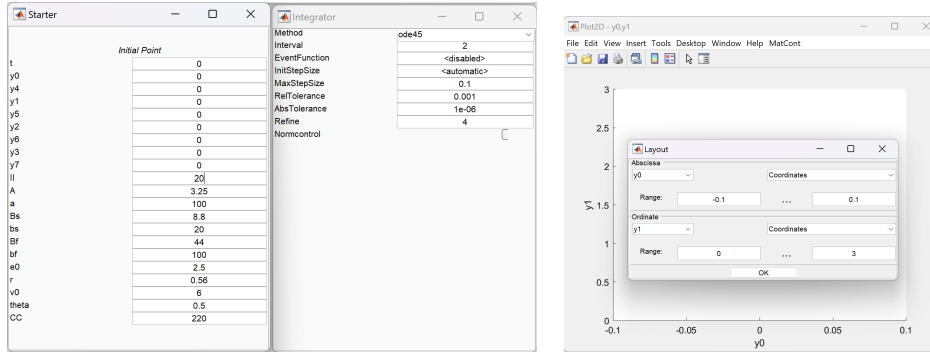


Figure 3: Specifying the parameter values and Layout of the Figure

We also want to see what the orbits look like, and for this, we open a 2D-figure `(MatCont):Window/Output|Graphic|2D plot`. In `Layout|Variables on axes` select  $y_0$  as abscissa(standard) and  $y_1$  as ordinate, and specify the ranges for  $y_0$  from -0.1 to 0.1 for the abscissa, and for  $y_1$  from 0 to 3 for the ordinate using `(Plot2D):MatCont|Layout`. Press 'OK'. In the 'Integrator' window, set the 'Interval' to 1 and 'MaxStepsize' to 0.1.

After these preparations, we are now ready to compute some orbits: Select `(MatCont): Compute|Forward`. Starting from  $\vec{y} = (y_0 \dots, y_7) = (0, \dots, 0)$ , the orbit converges to the point  $(.0014, 1.79, 4.24, 4.24, 0, 0, 0, 0)$ . We want to monitor this during the simulation via a Numeric window showing the current values of time, states and parameters, and (later in this tutorial) other quantities such as eigenvalues. Select `(MatCont): Window/Output|Numeric` and recompute the orbit. You can also try other initial points and check that these also converge to this point. So we have a(n apparent) global attractor. To see how  $y_0$  evolves over time, select `(2Dplot:1): Layout|Variables on axes` and select time and  $y_0$ . Redraw the curve via `(2Dplot:1): Plot|Redraw curve` and adjust the plot-ranges to obtain Figure 4(right).

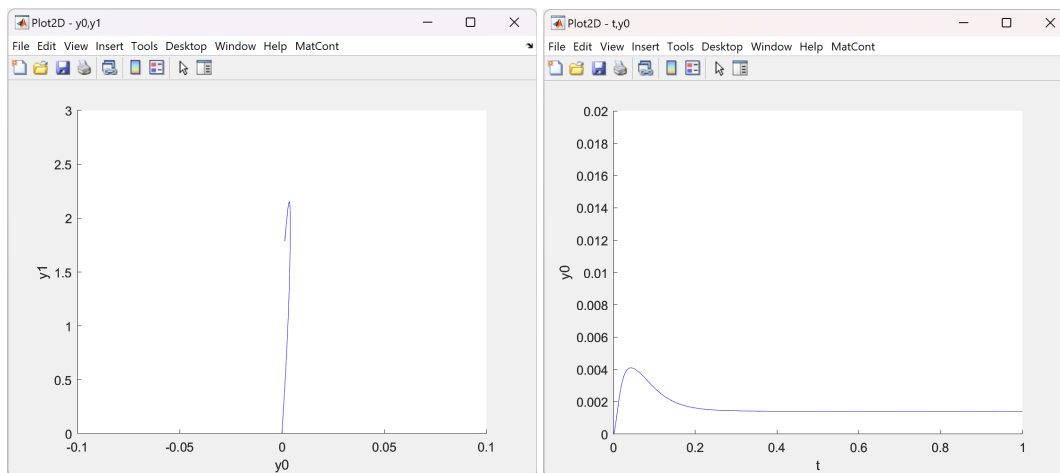


Figure 4: An orbit converging to a stable fixed point for  $CC = 220$ ,  $II = 20$ . Left:  $(y_0, y_1)$ -phase plane, Right: simulation showing  $y_0(t)$ .

Now change to  $II = 135$  and again compute an orbit via `(MatCont):Compute|Forward`. The result is quite different. To see the full range of  $y_0$ , we change the ranges for the coordinates. We do this by `MatCont|Fit Range`. As before, we could have used the `MatCont|Layout` option to change the plotting range. It is also possible to use the zoom option (the + in the Matlab figure) on the top right or to enter manually on the command line `y1im([0 0.14])`. The result will resemble Figure 5(right). We see some periodic

behaviour with a frequency of 3 Hz, and it looks like a spike-wave discharge pattern, as seen in EEG. Of note, it has some wiggle midway, and we will return to that later.

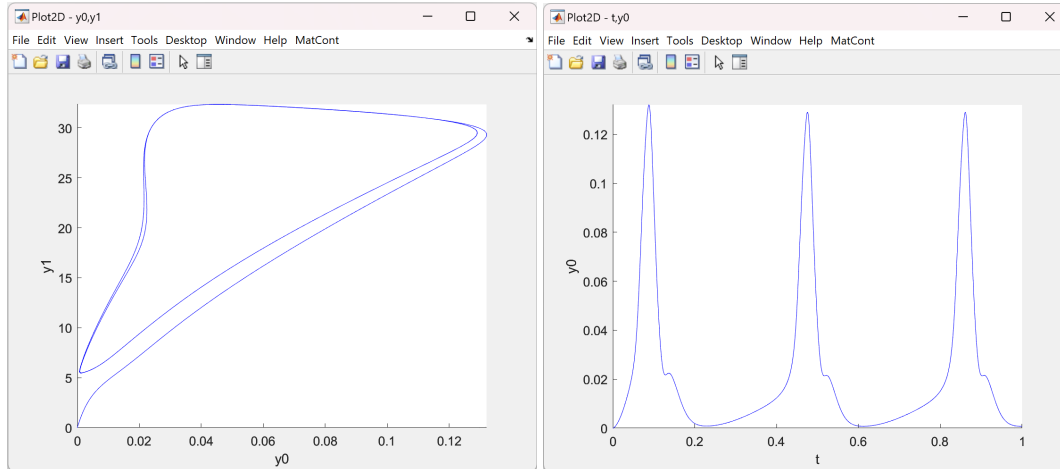


Figure 5: An orbit converging to a stable periodic orbit for  $CC = 220$ ,  $II = 135$ .  
Left:  $(y_0, y_1)$ -phase plane, Right: simulation showing  $y_0(t)$ .

### 3.3 Continuation of Equilibria in one parameter

We will now study how the stability of the equilibrium depends on the parameter  $II$ , i.e. the external background input. Select the last point via `(MatCont):Initial point` and then choose the last point on the previous curve. Change the curve type to EP\_EP by selecting `(MatCont):Type Initial point Equilibrium`. Activate the parameter  $II$  by clicking on the radio button next to it, and it will change colour to indicate this parameter is now selected to be varied. In the 2Dplot:1 window change the abscissa to the parameter  $II$  via `(2Dplot:1):Layout Variables on axes`. Also, change the plotting region for  $II$  to -250 to 300 and for  $y_0$  to 0 to 0.1. The standard values are fine, but to speed up the continuation, we will set `(Continuer):MaxStepsize=5` `(Continuer):MaxNumPoints=600`. Choosing good settings depends highly on experience.

The stability changes upon the variation of  $II$ . Thus, we would also like to see how the eigenvalues of the Jacobian matrix at the equilibrium develop. For this, we select `(Numeric):Layout` and highlight the option `Eigenvalues` (a checkbox appears in front of it). Note only the active parameters are shown. The Starter window shows the values of the other parameters. Press `OK`. Select `(MatCont): Compute Forward`.

**NOTE:** As we start the continuation, a new window appears: **Control**. This window is the only responsive panel during the computation. All other windows have greyed out as they are inactive; You cannot change settings during the computation. The Control panel reports the status "Computing..." and has two buttons 'Pause' and 'Stop'. Whenever a bifurcation or other special is detected, MatCont pauses, and the message changes to show some relevant information. The Pause-option will change to 'Resume' to continue the continuation. In the sequel, a few such points will be detected, but we will not always indicate to resume the computation. Once the computation has finished, the status will change to 'Finished', and the Pause option changes to 'View Result'. Selecting this option opens the Data Browser, where the result can be inspected in more detail.

At  $II = 109.388594$ , the message `Limit Point` appears in the status field of the MatCont window. Here, one eigenvalue has zero real part. You can check that in the numeric window! The curve (branch) of equilibria has a turning point. The equilibria were first stable and now turn unstable. Press the `Resume`

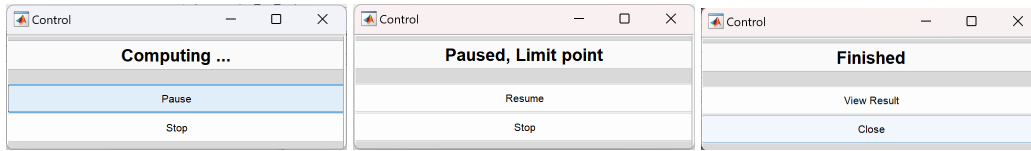


Figure 6: Control Panel during the continuation, at a special point, and after the computation.

button in the Control window to continue the computation of the currently unstable branch. In the meantime, MatCont finds a new special point, i.e. a test function is zero, but that is due to two real eigenvalues, one positive and one negative, summing up to zero, not a complex pair. This point is mentioned in the Control window as a Neutral Saddle. At  $II = -211.513481$ , another Limit Point is found. Here, the branch does not become stable; just one more eigenvalue has a positive real part as well. When the computation reaches  $II = -188.997488$ , two complex conjugate eigenvalues have zero real part, and a Hopf bifurcation is detected. The Plot2D figure will now look like Figure 7.

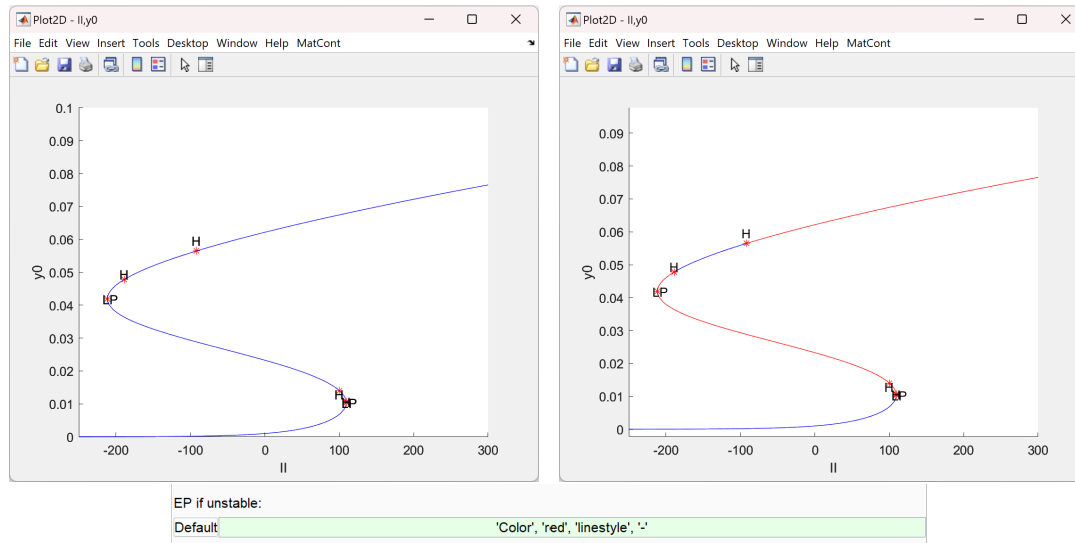


Figure 7: Left: A branch of equilibria in the  $(II, y_0)$ -plane displaying co-existence, Limit Point and Hopf bifurcations. Right: Same plot except the blue and red indicate which part of the branch of equilibria is stable and unstable, respectively. Bottom: Plot option to highlight the unstable part with a different colour.

If you feel the computation is too fast to follow, you can select to pause the computation of the EP\_EP curve after each computed point. For this choose `(MatCont):Options|Pause|At each point`. (Do not forget to set it back to 'At special points' after this computation.

Now we turn to the command line, i.e. the Matlab window, where you most probably now have the following information:

```

first point found
tangent vector to first point found
label = LP, x = ( 0.010348 -0.000000 6.716992 0.000000 5.517090 0.000000 5.517090 0.000000 109.388594 )
a=6.161978e+01
Neutral saddle
label = H , x = ( 0.010630 0.000000 6.814012 0.000000 5.562901 0.000000 5.562901 0.000000 109.333543 )
Neutral saddle
label = H , x = ( 0.014027 -0.000000 7.932057 0.000000 6.145333 0.000000 6.145333 0.000000 100.121277 )
label = LP, x = ( 0.041898 0.000000 17.675297 0.000000 13.563251 0.000000 13.563251 0.000000 -211.513481 )
a=2.696137e+00
label = H , x = ( 0.047787 -0.000000 20.342252 0.000000 15.905967 0.000000 15.905967 0.000000 -188.997477 )

```

```

First Lyapunov coefficient = 7.016601e-05
label = H , x = ( 0.056467 -0.000000 24.850742 0.000000 19.975932 0.000000 19.975932 0.000000 -91.880624 )
First Lyapunov coefficient = -2.370306e-04
elapsed time = 94.5 secs
npoints curve = 600

```

The first two lines indicate that the continuation has successfully started, and the last two lines show that it has ended, mentioning the number of points computed and the duration (including waiting time). Any other messages are displayed here, for instance, the four bifurcations together with their normal form coefficients. This information is useful as it indicates the non-degeneracy and criticality of the bifurcations. First, the two Limit Point bifurcations are shown;  $x$  denotes the eight state variables  $\vec{y}$  followed by the value of the active parameter  $II$  at the bifurcation point. The following line provides the normal form coefficient. Note that for a Limit Point, the sign of this coefficient is not unique<sup>2</sup>. Finally, we know that the limit cycle born from the first and second Hopf bifurcation are unstable and stable as the first Lyapunov coefficient is positive and negative, respectively.

Figure 7 shows a phenomenon called hysteresis or bistability. This phenomenon is an important feature of nonlinear (differential) equations. However, the plot does not yet indicate the stability. We can achieve this by selecting '**MatCont**:**Options**|**Plot Properties**'. In the Plot2D, select **MatCont**|**Redraw diagram**. You will now see the plot as in Figure 7 (but with vertical stripes from the simulations). Typically, we would expect the upper part to the right of the last Hopf to be stable, but that is not the case. Apparently, there will be some other attractor, as the simulation for  $II = 135$  already suggests.

For further reference, we will rename this curve so it is not overwritten. After the continuation has finished, press **View Result**. The Data Browser opens, and we see the data of the current curve. To go up one level, at the top, click on **Diagram; diagram**. Now, the option to rename the curve appears. Select the computed curve EP\_EP(1) a different name, say 'steadystates'.

### 3.4 Continuation of Bifurcations of Equilibria in one parameter

The bistability in Figure 7 is usually delimited by a wedge of Limit Point<sup>3</sup> bifurcations in parameter space. We compute this wedge here.

## Computing a Limit Point curve

Select one of the Limit Points found in the equilibrium continuation as the initial point '**MatCont**:**Select**|**Initial point**'. The Data Browser opens, and you can choose one of the special points as the initial point. Activate  $II$  and  $CC$  as active parameters, and set the MaxStepsize to 1.0. Change the ordinate of the 2D plot to the parameter  $CC$  and its range from 0 to 250. Now select '**MatCont**:**Compute**|**Forward**' and when finished also '**Backward**'. At  $(CC, II) \approx (59.11, 168.7)$  a Cusp bifurcation (CP) and at  $(II, CC) = (1.6840517, 1.6829686)$  a spurious Zero-Hopf bifurcation is detected. You probably need to enlarge the curve using '**MatCont**:**Compute**|**Extend**' to reach the plot boundary. You have now computed the wedge, see Figure 8, within which you have bistability. This figure is not the full story as we have already found Hopf bifurcations, implying that it is more co-existence and not bistability of equilibria.

## Computing a Hopf curve

To compute a curve of Hopf bifurcations, select a Hopf point analogously to the Limit point as the initial point. The default option is to go for a limit cycle. Change the curve type to H\_H via either the dropdown menu next to the initializer in the main window or '**MatCont**:**Type**|**Curve**|**Hopf**'. Again, use  $II$  and  $CC$  as active parameters. Press '**Compute**|**Forward**' and '**Compute**|**Backward**' to obtain the diagram as in Figure 9.

<sup>2</sup>This depends on a choice for an eigenvector for  $Aq = 0$ , where both  $q$  and  $-q$  are proper, but yield opposite signs for the normal form coefficient. At least, it is nonzero, hence non-degenerate.

<sup>3</sup>Be aware that Limit Point, Saddle-node and Fold all refer to the same bifurcation. Within a continuation context, Limit Point is a more general term as it could refer to any system of equations.



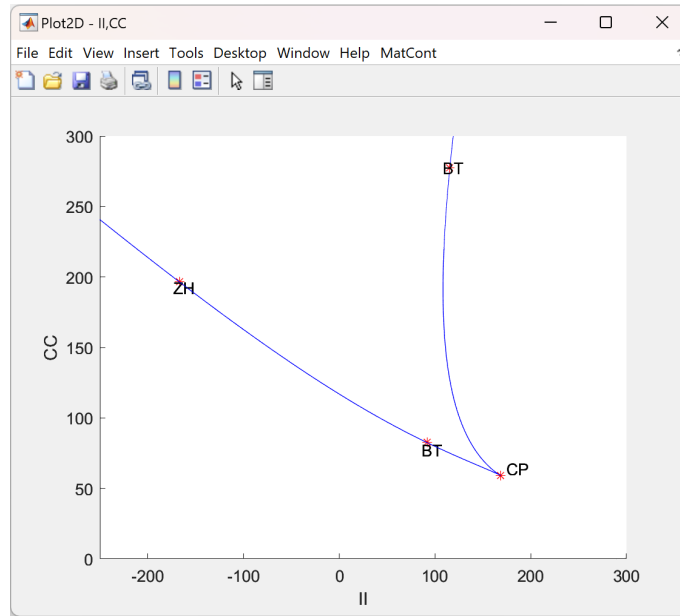


Figure 8: Limit Point bifurcation curves in the  $(II, CC)$ -plane with codimension 2 points with a wedge of co-existence of equilibria.

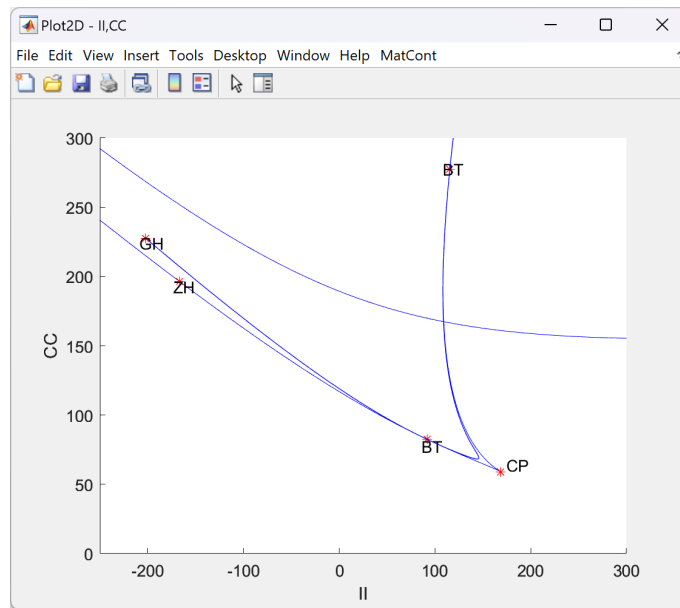


Figure 9: Limit Point and Hopf bifurcation curves in the  $(II, CC)$ -plane with codimension 2 points. The ZH is spurious, while the GH indicates where the Hopf bifurcation changes from sub- to supercritical. The Hopf curve emerges from a Bogdanov-Takens point.

## 4 Continuation of Limit Cycles

### 4.1 Starting from a Hopf point

We will now compute a branch of limit cycles in two ways, starting first from the Hopf point and second from a simulation. Select one of the Hopf points of the curve 'steadystates' using '(MatCont):**Select Initial point**' as the new initial point. Change the Initializer to H\_LC. Activate the parameter  $II$  and the period by clicking the radio buttons in the *Starter* window, and check that the box for detecting Limit Point of Cycles is ticked. To speed up computations, we will set MaxStepsize=100. This value is much higher than you would often use, but for this tutorial, this is acceptable. Close the 2DPlot (for speed reasons: plotting the cycles takes too much time now) and '(MatCont):**Compute Forward**'.

At  $II = 60.61$ , we see a Limit Point of Cycles bifurcation, and the periodic orbit becomes unstable. Resume the computation and stop when you are near  $II = -188$ . Here the branch terminates in the subcritical Hopf bifurcation. Open a 3D plot with '(MatCont):**Window/Output Graphic 3D Plot**'. Choose Abscissa= $z$ , Ordinate= $F$ , Applicata= $y$  in the Layout window. Change the plotting the region to  $II \in [-200, 120]$ ,  $y_0 \in [0, 0.15]$ ,  $y_1 \in [0, 35]$ . Alternatively, first redraw the curve, and then select Fit Range. Your 3D plot will resemble the one in Figure 10.

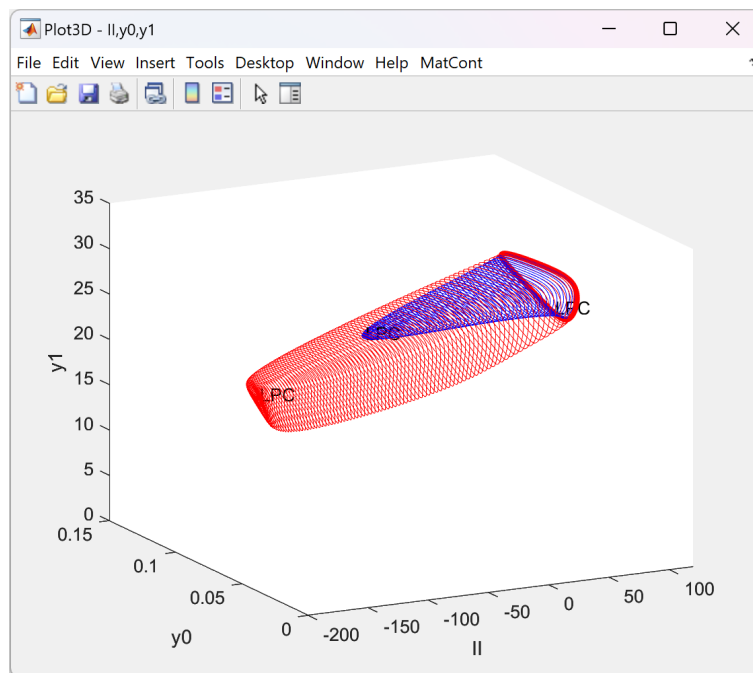


Figure 10: Equilibria and Limit cycles in  $(II, y_0, y_1)$ -space. The points and limit cycles which undergo a bifurcation are indicated with red dots. The blue part corresponds to stable limit cycles, and the red to unstable. At both ends, this branch emerges from a Hopf point.

### 4.2 Starting Limit Cycle from a time simulation

The continuation starting from the Hopf bifurcation worked, but it takes forever before the branch of periodic orbits comes to a relevant parameter regime, i.e.  $II > 70$ . In fact, it did not even get there at all. On the other hand, we have already found an interesting periodic orbit with simulations, and we will use that one to start the continuation directly in the more interesting parameter region. This method takes a simulation close to the periodic orbit and covers it a little more than one period. Then, starting from the last point, it looks at the first point backwards that is close to it. If it finds one, it takes the time between these points

as a guess for the period and fits a mesh to the computed simulation. Often, this is good enough to start the continuation.

We will first recompute the simulation with periodic dynamics. Take  $\vec{y} = 0$  as the initial condition and set  $II = 135$ . Set the point type **MatCont:Type|Point** and press **MatCont:Compute|Forward**. Select the Last Point as the new initial point. The option Select Cycle is also present, but that will fail as the current data is not good enough as it contains transients and several periods. After having selected the last point of the previous simulation, we set in the *Integrator* window the interval to 0.4, which is an appropriate upper limit of the period. Press **MatCont:Compute|Forward**. Select **Plot2D:MatCont|Clear** and **Plot2D:MatCont|Redraw curve** to see that it is a periodic orbit.

Go to the Data Browser via 'View Result' and now select the option **Select Cycle** button in the top left, a. A pop-up will appear with two fields: "Tolerance" equal to 1e-2 and "test intervals" equal to 40. These values are adequate for this example, so press **OK**. We select  $II$  as a free parameter in the new *Starter* window. As the period changes along the branch, we need to set the period as a free parameter as well. Change **MaxStepsize** to 200 in the *Continuer* window, and open a **Window/Output|Numeric** window. *Optional:* Close the figure, as plotting it during the continuation slows it down, and re-open it later. Press **MatCont:Compute|Forward**. Most likely, you will notice in the numeric window that  $II$  decreases while the period goes up. Around  $II \approx 110$ , the period still goes up, but  $II$  only slightly decreases. Here, the periodic orbit gets close to a homoclinic orbit, and it cannot be continued further. Rename the computed curve to LC\_part1.

Before we compute the other part of the branch, check that under Singularities, the occurrence of a Limit Point of Cycles is monitored in the starter window. Press **MatCont:Compute|Backward**. You will observe two LPC bifurcations at  $II = 169.7$  and  $II = 162.2$ . We can also inspect the Floquet multipliers to observe a double multiplier +1. We conclude this is again a region of bistability. Hereafter, the continuation will continue indefinitely, and  $II$  will keep increasing, so you can stop it. Rename the computed curve to LC\_part2.

### 4.3 Continuation of LPC in two parameters

Similar to the fold and Hopf bifurcations of equilibria, we want to compute the curves in parameter space where the dynamics change. Select one of the computed LPC points as initial point '**MatCont:Select|Initial Point**'. Next, set  $II$  and  $CC$  as active parameters and '**MatCont:Compute|Forward**'. You can monitor the result via the Numeric window or a 2D Plot. You can add a plot of this curve to the bifurcation diagram shown in Figure 9. Does the LPC end at a Generalized Hopf bifurcation, as expected?

## 5 Visualizing your results

We have already plotted some of the computations. It is also possible to load the output and plot and manipulate it yourself. We will give a nontrivial example of plotting the profile. First, we must access the computed curves and load the data into the Matlab workspace. In the MatCont folder, there is a subfolder called 'Systems/JR\_slow/diagram'. This folder contains the data of the curves as files such as 'LC\_part1.mat'. Load the curve into the Matlab workspace with the command: `load Systems/JR_slow/diagram/LC_Part1.mat`. Your workspace now contains five variables  $x, v, h, f, s$  and several more. They contain the following data

- x Array containing the phase variables and parameters.
- v Array containing the tangent vector.
- h Array containing the step size, Newton steps and values of test functions.
- f Array containing eigenvalues or Floquet multipliers and the time discretization.
- s A structure specifying the special points, i.e. detected bifurcations and the first and last point.

## 5.1 Plotting the profile

Plotting the profile, i.e. the evolution of a variable during the whole period, is nontrivial as it is stored in a condensed way. Suppose we know how many mesh points were used. Above, we use  $ntst = 40$ . Otherwise, we could look up the entry in the f-variable equal to 1. Now we need

- the index of the periodic orbit we want to plot: say  $ii$ .
- the profile of one variable, say the  $jj$ -th:  $x(jj:ndim:end-2,ii)$
- the time-mesh  $\tau=f(1:ntst+1,ii)$ , with  $0 = \tau_0 < \tau_1 < \dots < \tau_{ntst} = 1$ .
- the period  $T = x(end-1,ii)$  to scale the mesh back from  $[0,1]$  to  $[0,T]$

The f-variable stores only the mesh points  $t_i$ . On each interval, the orbit is also computed at equidistant time-points  $t_{i,j} = t_i + \frac{j}{m}(\tau_{i+1} - \tau)$ ,  $j = 0, 1, \dots, m$ . We can use this detailed grid for a nicer, smoother plot. We have the following code:

```
ii=10; % The i-th point on the branch
jj=1;  % The j-th variable that we're going to plot.
ndim=8;% The number of state variables
ntst=40;ncol=4; %From the continuation
mesh=sort([reshape(repmat(f(1:ntst,ii),1,ncol)+...
    repmat((0:ncol-1)/ncol,ntst,1).*repmat(diff(f(1:ntst+1,ii)),1,ncol),ntst*ncol,1); 1]);
figure;plot(mesh*x(end-1,ii),x(jj:ndim:end-2,ii))
```

## 5.2 Identifying Spike Wave Discharges

One particular feature of this model is that it has Spike-Wave Dynamics. That is, on top of the 3Hz wave, there is an additional spike that can be detected as a local extremum. The stability of the periodic orbit does not change, and there is no qualitative change in the dynamics. The appearance of such a local extremum is also called a 'ghost' or 'false' bifurcation. The code below illustrates these local extrema.

```
load Systems\JR_slow\diagram\LC_part1.mat
out=x(3:8:end-2,:)-(x(5:8:end-2,:)+x(7:8:end-2,:))/2;
extrema=nan(6,size(out,2));
for i=1:size(out,2)
    diff_tr=diff(out(1:end-1,i));
    diff_tr_shift=diff(out(2:end,i));
    temp=[out(find([ (diff_tr>0).*(diff_tr_shift<0) ]),i);...
    out(find([ (diff_tr<0).*(diff_tr_shift>0) ]),i)];
    extrema(1:size(temp,1),i)=temp;
end
figure(1);hold on;plot(x(end,:),sort(extrema));
```

Do this again for LC\_part2.mat, and your plot will look like Figure 11.

## 6 Tips

- From the graphical plots, you can click on the label of a bifurcation to directly load this point as the new initial point.
- One can right-click in the Main MatCont window. For the current system, options appear related to the system. For the current curve, computation and data options are available. This feature is helpful if some window is accidentally closed.
- If the results are good enough for publication, create a final figure with the raw data in Matlab or some other routines. The plots are good during the exploration of the system, but they are objects with lots of additional data. Hence, if you save these figures, the figure becomes unnecessarily large.

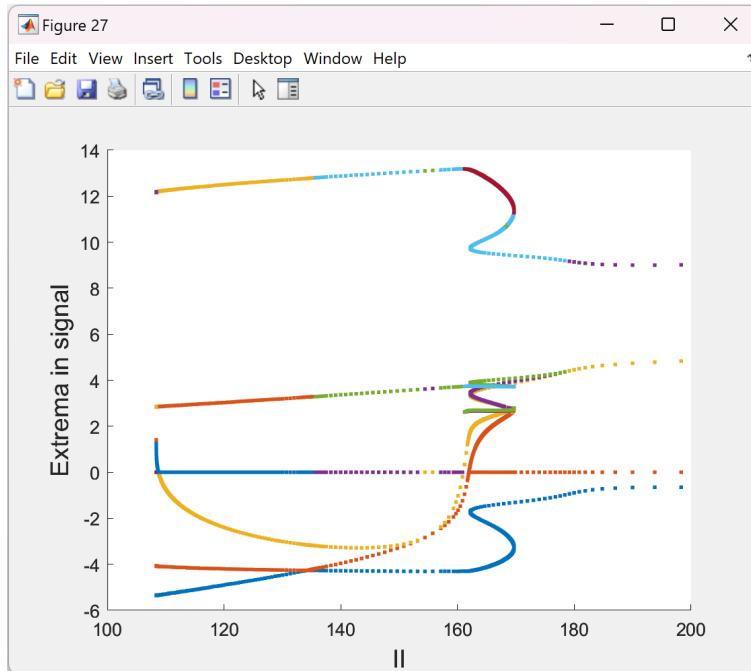


Figure 11: Extrema in the output signal. The appearance of local extrema indicates spikes in the waveform. Near the fold bifurcations, the resolution suffers from the large value for the step size.

## References

- [1] New features of the software MatCont for bifurcation analysis of dynamical systems. Dhooge, A. and Govaerts, W. and Kuznetsov, Yu.A. and Meijer, H.G.E. and Sautois B., *Mathematical and Computer Modelling of Dynamical Systems*, Vol. 14, No. 2, pp 147-175, 2008, DOI:<https://doi.org/10.1080/13873950701742754>.
- [2] Goodfellow M., Schindler K. and Baier G., Intermittent spike wave dynamics in a heterogeneous, spatially extended neural mass model, *NeuroImage* 2011, Vol 55(3), pp. 920-932, DOI: <https://doi.org/10.1016/j.neuroimage.2010.12.074>.
- [3] Govaerts, W. and Sautois, B., Phase Response Curves, Delays and Synchronization in Matlab, *Lecture Notes in Computer Science* 3992 (2006) 391-398, [https://link.springer.com/chapter/10.1007/11758525\\_53](https://link.springer.com/chapter/10.1007/11758525_53).
- [4] Homburg, A.J. and Sandstede, B.: Homoclinic and heteroclinic bifurcations in vector fields. *Handbook of Dynamical Systems III* (Edited by H Broer, F Takens and B Hasselblatt). Elsevier (2010) 379-524. DOI: [https://doi.org/10.1016/S1874-575X\(10\)00316-4](https://doi.org/10.1016/S1874-575X(10)00316-4), Also available at [http://www.bjornsandstede.com/papers/Survey\\_Homoclinic\\_Bifurcations.pdf](http://www.bjornsandstede.com/papers/Survey_Homoclinic_Bifurcations.pdf) (Retrieved 1-11-23).