1(a) Make sure that AUTO-07P is installed on your computer. Type `auto` to enter in the AUTO-shell, then open the manual using the command `mn()`.

It might be useful for you to print out the quick reference table for the meaning of the various AUTO constants in section 10 of the manual and the list of commands, explained in section 4.13 of the manual.

(b) In this session we are going to start by following periodic orbits in AUTO, this can be achieved with `IPS=2`. A good way to compute periodic solutions is often to compute steady states (`IPS=1`) and compute a periodic orbit starting at a Hopf bifurcation.

To familiarise yourself with this try running the auto demo `abc`. A useful tip: *always start a new demo by first moving to a clean directory.* So make a new directory using the command `mkdir abc`, then enter in that directory using `cd abc`. Copy here the files of the demos with the AUTO command `dm('abc')` and then follow the example using exactly the steps described in table table 14.7 of section 14.2 of the AUTO of section 14.2.

Plot the final output using `p2('abc')`. Use the interactive plotting commands to find out what the bifurcation diagram means. A useful tip at this stage is to clean the directory so that it just contains the required output files, via typing `cl()`

2 We will run the demo `pp2` which is a two-dimensional predator prey model. We are going to run the demo in two different ways. The first step in running a new AUTO demo is always to move to a clean directory. So type `cd ..` to come back to your home directory, and then `mkdir pp2` to make the directory and `cd pp2` to enter in it.

(a) The first command `dm('pp2')` will load the demo. That is, it will copy the files `pp2.f90` and `c.pp2` into your directory. Open these files and have a look at their content. Try to familiarise yourself with what each of the constants mean. The command

```
auto('pp2.auto')
```

will run the demo by first computing branches of equilibria then following a branch of periodic orbits born in a Hopf bifurcation.

Note how in the second run the period `PAR(11)` is included in the list `ICP` of active continuation parameters.

The manual then explains how to plot the data using `p2(pp2+ps)`. See how the periodic orbit approaches a heteroclinic orbit.

Play with the plotter to get different versions of the graphs. Try to get postscript output from the plotter and print the output.

(b) Now try repeating the above computations using the Python interface and typing the commands contained in `pp2.auto` directly in the auto shell. So clean the directory using the command `cl()`. The first command

```
pp2 = run(e='pp2',c='pp2')
```

will make the first run and store the solutions in the local variable `pp2`. We can save the data in three output files with the command `sv(pp2,'run1')` (that save the solutions stored in the variable `pp2` using as extension the given string `run1`). If you now watch at the files in the directory (using for example the command `ls`) you can find the files `s.run1` (solutions at labeled points) `b.` (bifurcation diagram) and `d.run1` (run-time diagnostics).

To make the second run we can follow two ways: the first one is edit the file `c.pp2` to change the AUTO constants to start continuation of periodic orbits at the label of the Hopf bifurcation

```
IPS=2   IRS=8   ICP = [1,11]    UZR= {-11:  36}
```

```
r(e='pp2',s='run1',c='pp2.1',sv='run2')
```

The second way is the one reported in the auto file, i.e. with the commands

```
solution = pp2('HB1')
ps=run(solution,IPS=2,ICP=[1,11],UZR={-11:36}).
```

With the first command we are selecting as starting solution the first labeled as `HB` that is in the local variable `pp2`. With the second command we start the continuation from that point: so the equation file and the constants file is the same used in the previous continuation. Then we have to modify the constant that are in `c.pp2` to do this new continuation. Notice that in this case we haven't modified the starting label constant `IRS`, since we are telling to AUTO to start form that solution.

3 We shall now extend the computation in the previous demo by taking `PAR(11)` out of the list of active continuation parameters and including `PAR(2)` instead so that an approximation to the heteroclinic orbit is followed in the two parameters `PAR(1)` and `PAR(2)`.

(a) Before doing this we need to increase the accuracy of the approximation of the heteroclinic orbit. To do so we increase `NTST` (the number of discretisation intervals) and so the second run is obtained with the command

```
ps=run(solution,IPS=2,ICP=[1,11],UZR={-11:100}, NTST=100).
```

(b) Restart from the label with `PAR(11)=100` and continue the orbit with this period in the two parameters `PAR(1)` and `PAR(2)` using the command

```
heter=run(ps('UZ1'),ICP=[1,2],UZR={-1:[0,1]}).
```

Compute in both directions (by changin the sign of `DS`) and append the data into the same variable with the command

```
heter=heter+run(ps('UZ1'),ICP=[1,2],UZR={-1:[0,1]},DS='-').
```

Use the command

```
heter=rl(heter)
   p2(heter)
```

to relabel the output (this avoids double use of the same solution label in a single file) and to see the bifurcation diagram and the solution obtained. What is the approximate relation between `PAR(1)` and `PAR(2)` when the heteroclinic orbit exists?

4 We are now going to run the auto demo `tor` which is the same set of equations as the the HomCont demo `cir` that we shall consider in the second practical (but with a different fixed value of `PAR(3)` γ. It represents the equations for the dynamics of the simple electronic circuit studied by Freire *et al.*

(a) Make the first run and save the data. Change the value of `NMX to 300` in `c.cir.2` which contains the constants for the second run. The second run should then produce a periodic orbit that approaches a homoclinic orbit. Which equilibrium is the orbit homoclinic to?

(b) Add a function to `UZR` to compute detect the point where `PAR(11)=100`. Then orbits with fixed period 100 from this point by allowing the two parameters `PAR(1)` and `PAR(2)` to vary.

(c) Now do something similar for the third run of the demo by changing `NMX` to 300 in `c.cir.3` and finding a different homoclinic orbit. What is different about this homoclinic orbit? What is different about its path as the two parameters vary?

(e) (**optional**) Try continuing some of the `LP` points in two parameters that approach the homoclinic orbit that was computed in the part (c). Plot them in the same bifurcation daigram as the homoclinic orbit itself. What do you see?

5 We are now going to try our first HomCont demo. This is the demo `rev` which considers how to continue symmetric homoclinic orbits of a reversible fourth-order differential equation. We shall find out the details of how HomCont works in the next practical session.

(a) Load the demo `rev` as described in Chapter 26 of the AUTO manual. Note that for the reversible case, AUTO only computes the orbit on the half-interval up to its point of symmetry. This demo starts from data saved in a data file that was obtained by numerical integration; `rev.dat.1` and `rev.dat.3` which are symmetric under different reversibilities. Run the demo.

(b) Edit `rev.f`. Introduce two new parameters into the model so that the equation reads $u^{iv}(x) + Pu'' + u - (1+\alpha)u^3 + \beta u^5 = 0$. Do continuation until $\beta = 0.2$ and recompute the bifurcation diagrams for various values of α. Can you find examples of the 'homoclinic snake' presented in the lecture?

(c) Try writing a small utility programme (in *matlab* or *fortran*, *C* or whatever) that will read the data in from `rev.dat.*` and "double the data' to plot it on the full time interval. This can be achieved by setting $U(tmax + t) = RU(tmax - t)$ where $R = [1, -1, 1, -1]^T$ in the case of `rev.dat.1` and $R = [-1, 1, -1, 1]^T$ for `rev.dat.3`.

Plot the resulting data using a suitable plotting programme such as *matlab* or *gnuplot* to check the data is in the right format. We will use this data in the second computer practical.