

ME 406

Trapping a Limit Cycle

```
In[322] := sysid
```

```
Mathematica 4.1, DynPac 10.65, 2/13/2002
```

```
In[323] := intreset;
```

```
In[324] := plotreset;
```

■ Introduction

In this notebook, we localize a limit cycle by orbit trapping. The example we consider, given below, is taken from **Nonlinear Differential Equations and Dynamical Systems**, 2nd edition, Ferdinand Verhulst, Springer, 1996, example 4.8, page 48, with a correction of a sign error.

$$\dot{x} = y - x(x^2 + y^2 - 2x - 3), \quad \dot{y} = -x - y(x^2 + y^2 - 2x - 3).$$

We now define the system for *Mathematica*. We include a parameter a so that variations of the system can be studied easily. For our system, $a = 1$.

```
In[325] := setstate[{x, y}];
```

```
In[326] := slopevec = {y - a * x * (x^2 + y^2 - 2 * x - 3), -x - a * y * (x^2 + y^2 - 2 * x - 3)};
```

```
In[327] := setparm[{a}];
```

```
In[328] := parmval = {1};
```

```
In[329] := sysname = "LimCyc 4.8";
```

```
In[330] := plrange = {{-5, 5}, {-5, 5}};
```

```
In[331] := labshift = 10;
```

■ Equilibrium Point

We look for equilibrium states using findpolyeq.

```
In[332] := findpolyeq
```

```
Out[332] = {{0, 0}, {-i - 3 a / (2 a), (1 - 3 i a) / (2 a)}, {i - 3 a / (2 a), (1 + 3 i a) / (2 a)}}
```

Mathematica has found three equilibria of which only one is real. As shown in class, there are no other real equilibria. (Important caution: we never know for sure without independent confirmation whether a root-finder such as `findpolyeq` has found all of the equilibrium states.) Let's look at the stability of the equilibrium at the origin:

```
In[333] := classify2D[{0, 0}]

Abbreviations used in classify2D.

L = linear, NL = nonlinear, R2 = repeated root.

Z1 = one zero root, Z2 = two zero roots.

This message printed once.

unstable - spiral
```

This is typical of what we would find inside a limit cycle. In a search for periodic solutions, we move next to the Bendixson test.

■ Application of Bendixson Test

From the Bendixson test, we know that periodic solutions can occur only in regions where the divergence of the slope vector changes sign or is identically zero. We use the function `signcontour` to map the algebraic sign of the divergence here. The function `dival` returns the divergence with current values of the parameters substituted.

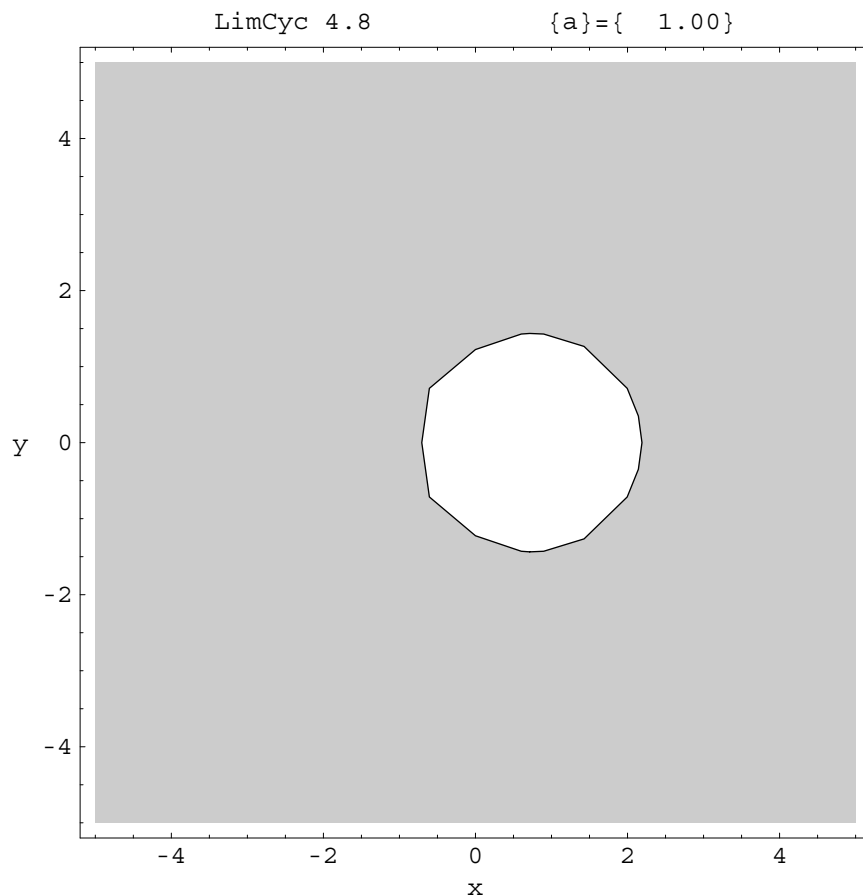
```
In[334] := bendix = dival[slopevec]

Out[334] = -x (-2 + 2 x) - 2 y2 - 2 (-3 - 2 x + x2 + y2)
```

We turn off the graphics axes and turn on the frame.

```
In[335] := axon = False; frameon = True;
```

```
In[336] := graph1 = signcontour[bendix];
```



The divergence is positive in the white region (which can be shown to be a circle with center at $x = 3/4$, $y = 0$ and radius $\sqrt{33}/4$). By the Bendixson criterion, any periodic solution will have to enclose both white and grey regions. This is only a necessary condition, and we now move to orbit trapping techniques to establish that there is a periodic solution.

■ Orbit Trapping

Because they are easy, we will start with circles. We define our trapping curve as a circle of radius b with center at the origin. We first define an arc and then a curve from the arc.

```
In[337] := trapperarc[b_] := {{b * Cos[u], b * Sin[u]}, {u, 0, 2 * Pi}};
```

```
In[338] := trappercurve[b_] := {trapperarc[b]}
```

Now we use the routine orbrcross to see whether a given circle traps orbits. We start with a radius $b = 0.5$.

```
In[339] := orbrcross[trappercurve[0.5]]
```

Tangencies: No.

Crossings in negative N direction: No.

Crossings in positive N direction: Yes.

Thus the circle with radius 0.5 traps orbits outside. If we can find a larger circle which traps orbits inside, we will have established the existence of a periodic solution. We try $b = 5.0$.

```
In[340] := orbcross[trappercurve[5.0]]
```

```
Tangencies: No.
```

```
Crossings in negative N direction: Yes.
```

```
Crossings in positive N direction: No.
```

We have been lucky. We now know that there is a periodic solution somewhere between $b = 0.5$ and $b = 5.0$. Let's refine our guesses.

```
In[341] := orbcross[trappercurve[1.0]]
```

```
Tangencies: Yes.
```

```
Crossings in negative N direction: No.
```

```
Crossings in positive N direction: Yes.
```

This is optimum because it still traps, but there are tangencies. For a slightly larger b , we would get crossings in both directions. Let's verify that:

```
In[342] := orbcross[trappercurve[1.1]]
```

```
Tangencies: No.
```

```
Crossings in negative N direction: Yes.
```

```
Crossings in positive N direction: Yes.
```

Now we locate more precisely an outer trapping curve, working back from $b = 5.0$.

```
In[343] := orbcross[trappercurve[4.0]]
```

```
Tangencies: No.
```

```
Crossings in negative N direction: Yes.
```

```
Crossings in positive N direction: No.
```

```
In[344] := orbcross[trappercurve[3.0]]
```

```
Tangencies: Yes.
```

```
Crossings in negative N direction: Yes.
```

```
Crossings in positive N direction: No.
```

This also appears optimum, and we verify it by a slightly smaller b :

```
In[345] := orbcross[trappercurve[2.9]]
```

```
Tangencies: No.
```

```
Crossings in negative N direction: Yes.
```

```
Crossings in positive N direction: Yes.
```

Now we have a periodic orbit trapped between $b = 1.0$ and $b = 3.0$. Let's plot these two curves in red:

```
In[346] := display = False;
```

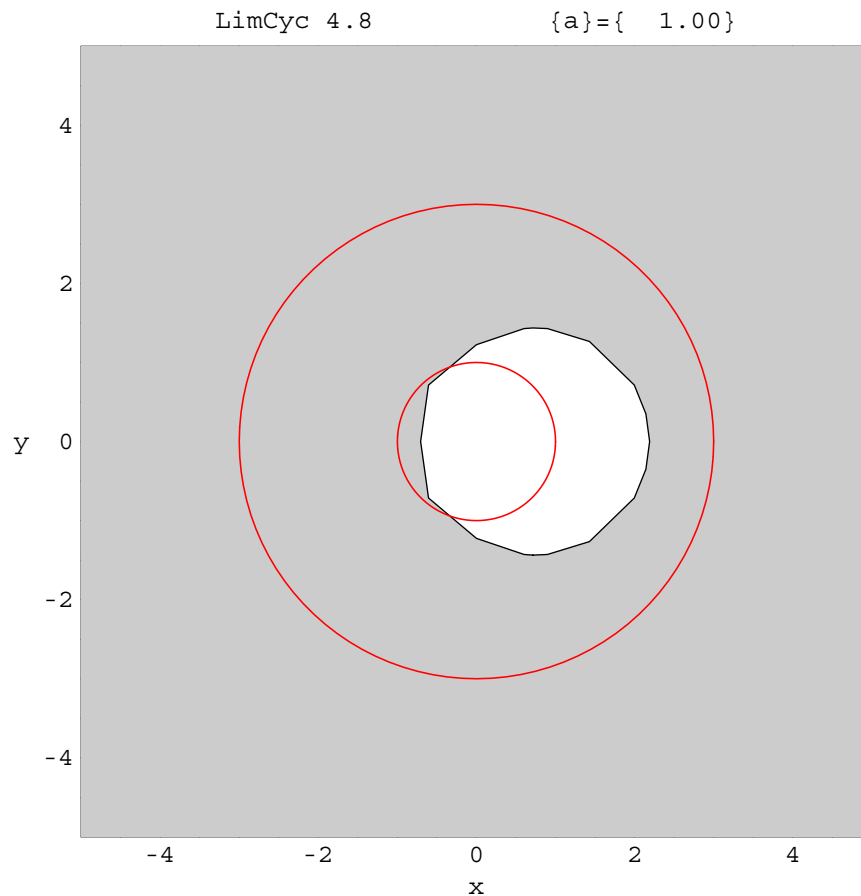
```
In[347] := setcolor[{Red}];
```

```
In[348] := graph2 = plotcurve[trappercurve[1.0]]; 
```

```
In[349] := graph3 = plotcurve[trappercurve[3.0]]; 
```

```
In[350] := display = True;
```

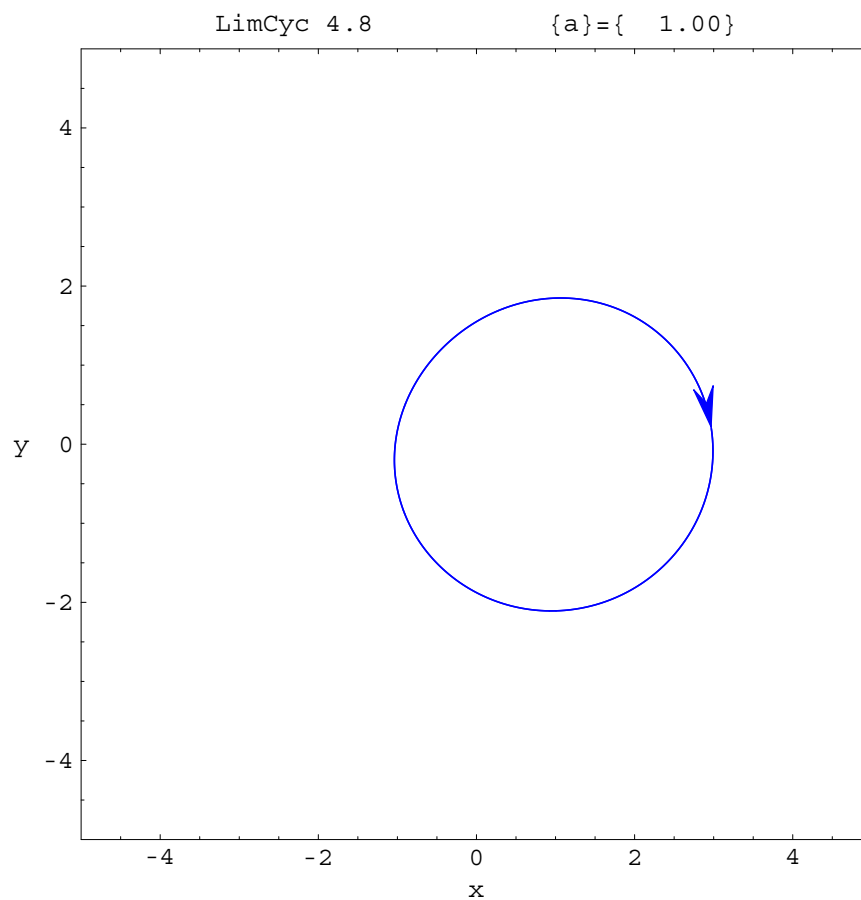
```
In[352] := graph4 = show[graph1, graph3, graph2];
```



Now let's add the limit cycle in blue.

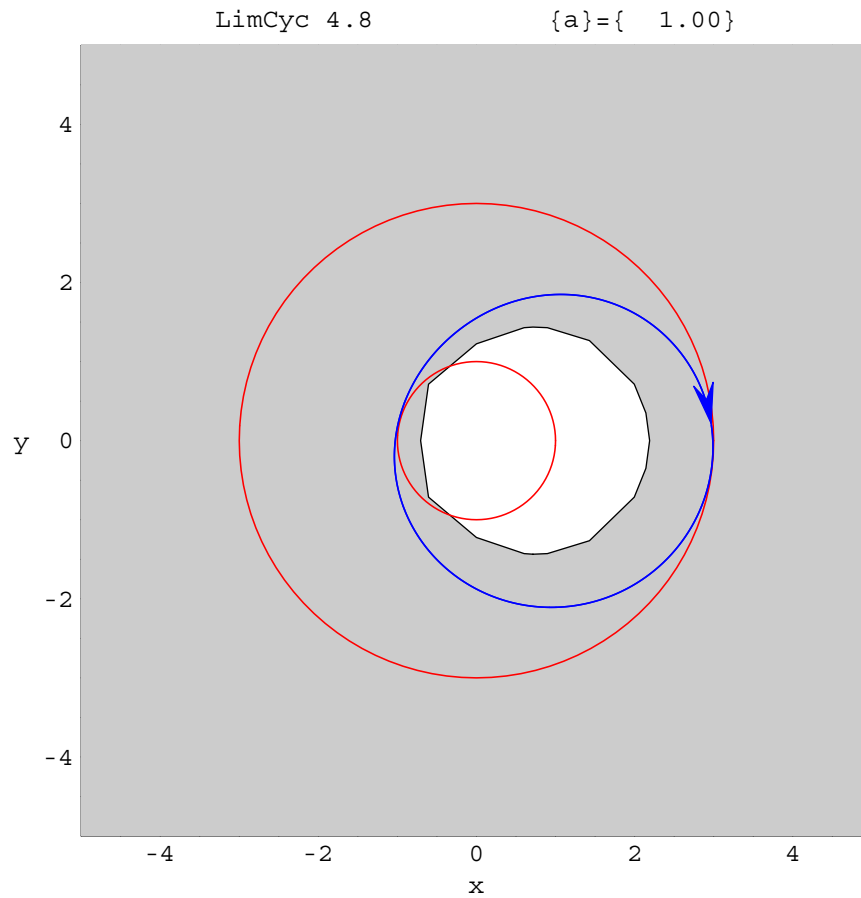
```
In[353] := nsteps = 500;
```

```
In[354] := h = 0.025;  
In[355] := t0 = 0;  
In[356] := initvec = {2, 0};  
In[357] := sol1 = limcyc[initvec, t0, h, nsteps];  
In[358] := arrowflag = True;  
In[359] := arrowvec = {1 / 2};  
In[360] := setcolor[{Blue}];  
In[361] := graph5 = phaser[sol1];
```



Now we put it all together:

```
In[362] := graph6 = show[graph4, graph5];
```



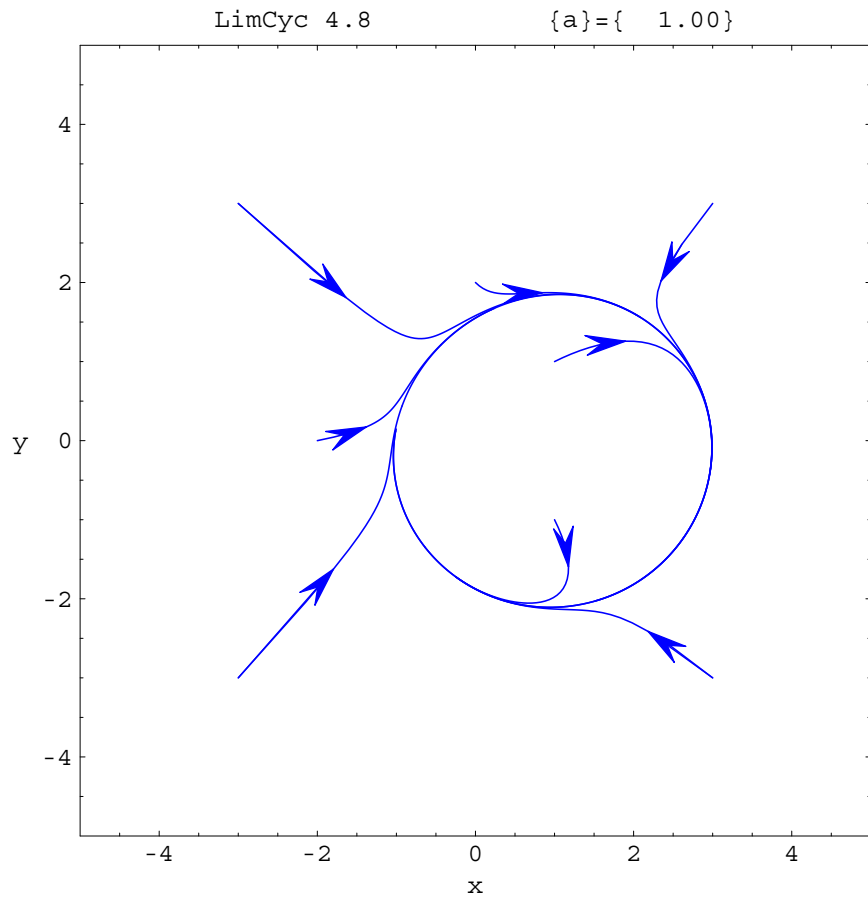
As a final calculation, we add a few orbits approaching the limit cycle, also in blue.

```
In[363] := initset = {{-2, 0}, {0, 2}, {3, 3}, {-3, -3}, {3, -3}, {-3, 3}, {1, 1}, {1, -1}};
```

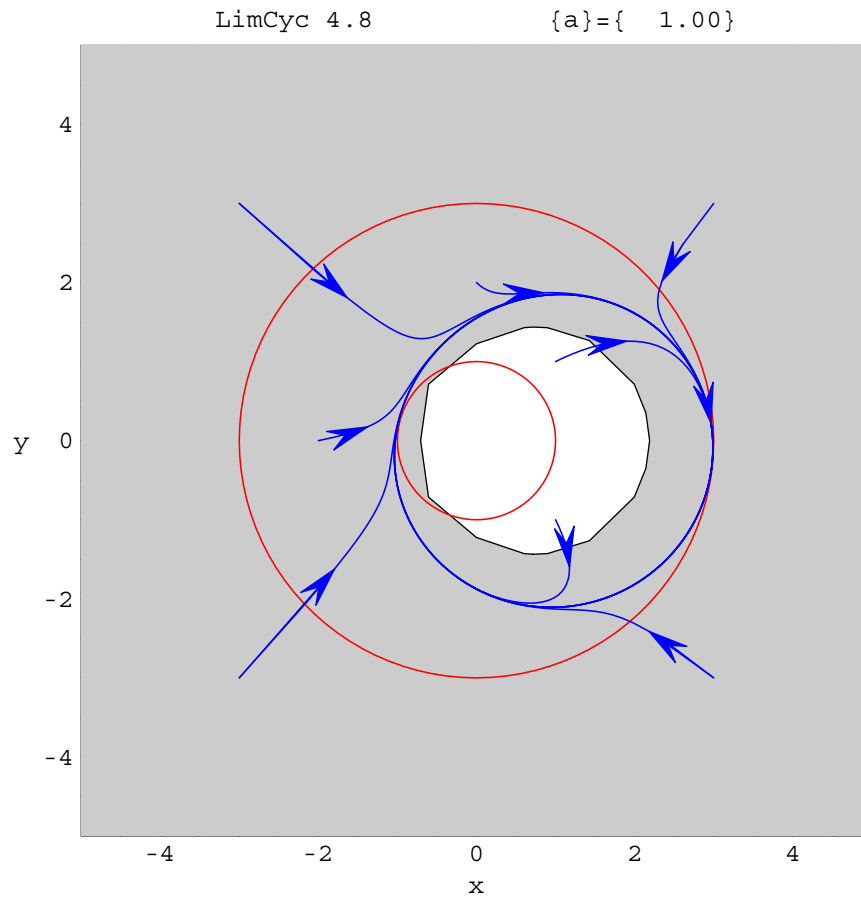
```
In[364] := nsteps = 100;
```

```
In[365] := arrowvec = {1 / 8};
```

```
In[366]:= graph7 = portrait[initset, t0, h, nsteps, 1, 2];
```



```
In[367]:= graph8 = show[graph6, graph7];
```



The approach orbits show clearly the stability of the limit cycle.