ME 406 A Graphical View of the Transition from Local to Global

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

```
Mathematica 4.1, DynPac 10.65, 2/4/2002
```

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

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

■ INTRODUCTION

In this notebook, we study the transition between a local and global description of a saddle point. We construct a sequence of graphs which can be animated to give a movie of the transition. This is done for a particular system, but the code could be applied to any phase portrait.

■ DEFINING THE SYSTEM

The system we use here was given as an example in **Nonlinear Ordinary Differential Equations**, D.W. Jordan and P. Smith, second edition, Oxford Press, 1987, pp. 51-52.

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

```
In[326]:= setparm[{}];
```

```
In[327]:= parmval = {};
```

```
In[328]:= slopevec = {x - y, 1 - x * y};
```

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

■ EQUILIBRIUM POINTS

We use findpolyeq to find the equilibrium points:

```
In[330]:= eqpoints = findpolyeq

Out[330]= {{-1, -1}, {1, 1}}
```

```
In[331]:= eq1 = eqpoints[[1]]

Out[331]= {-1, -1}
```

```
In[332]:= eq2 = eqpoints[[2]]

Out[332]= {1, 1}
```

We now classify these:

In[333]:= **classify2D[eq1]**

```
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
```

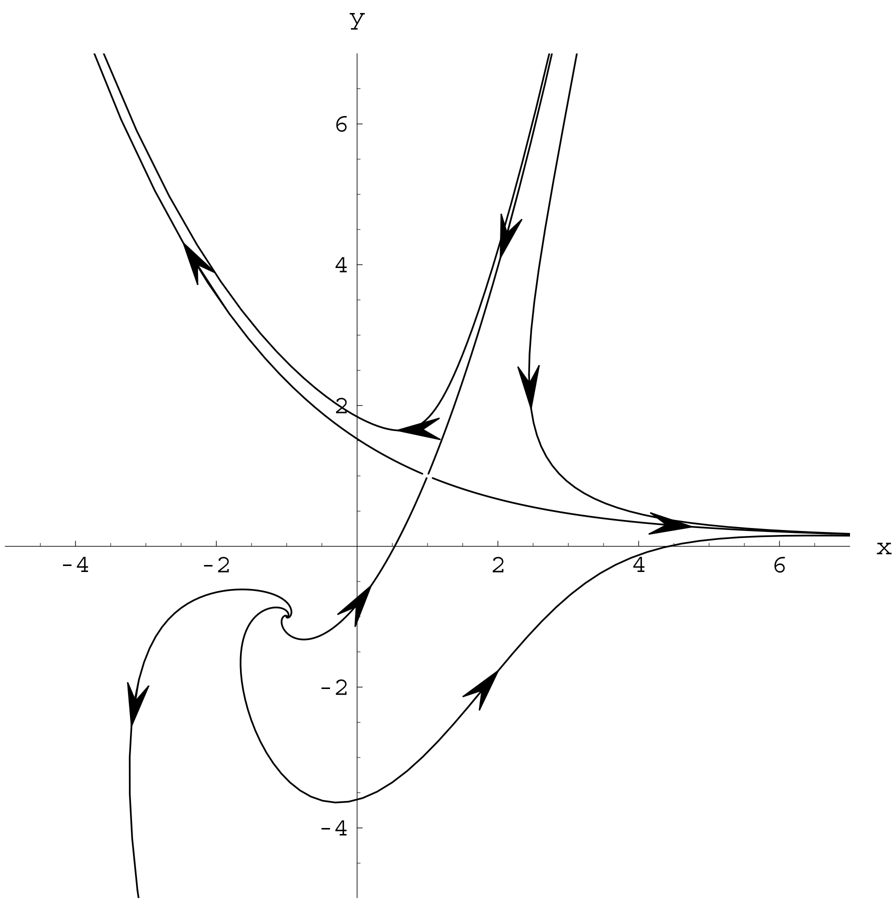
In[334]:= **classify2D[eq2]**

```
unstable - saddle
```

Thus we have an unstable spiral and an unstable saddle. We use saddleportrait to get a quick look at the phase portrait.

In[335]:= **plrange = {{-5, 7}, {-5, 7}};**

In[336]:= **quickgraph = saddleportrait[eq2, plrange];**



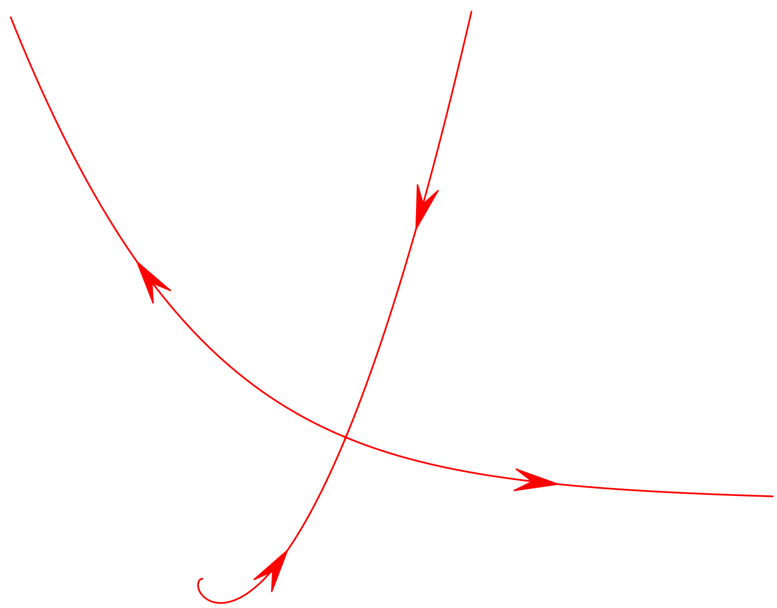
This shows the basic topology of the orbits and the connected unstable spiral and saddle. We will now construct more carefully a phase portrait which will then be the basis of our movie. We could of course focus on either the spiral or the saddle point. We choose to focus on the saddle point here. We start with the stable and unstable manifolds of the saddle point, with the curves being shown in red. We construct these manually so that we have full control over the time step and starting points. We start by getting the eigenvectors and eigenvalues at the saddle point.

```
In[337]:= eiger = Eigensystem[dermatval[N[eq2]]]

Out[337]= {{-1.41421, 1.41421}, {{0.382683, 0.92388}, {0.92388, -0.382683}}}

In[338]:= vec1 = First[Last[eiger]]

Out[338]= {0.382683, 0.92388}

In[339]:= vec2 = Last[Last[eiger]]

Out[339]= {0.92388, -0.382683}
```

Now we will construct the integral curves entering and leaving the saddle. We do this by choosing initial conditions displaced from the equilibrium a small distance along the eigenvectors. We choose the time direction of integration to move away from the point on all four curves. We also use range checking on the integration to prevent overflow. We turn arrows on, and put an arrow at the midpoint of each curve. We set the color to Red. We turn the axes off.

```
In[340]:= eps = 0.001;

In[341]:= rangeflag = True;

In[342]:= ranger = plrange;

In[343]:= arrowflag = True;

In[344]:= arrowvec = {1 / 2};

In[345]:= setcolor[{Red}];

In[346]:= axon = False;

In[347]:= sol1 = integrate[eq2 + eps * vec2, 0.0, 0.005, 2000];

In[348]:= display = False;

In[349]:= man1 = phaser[sol1];

In[350]:= sol2 = integrate[eq2 - eps * vec2, 0.0, 0.005, 2000];

In[351]:= man2 = phaser[sol2];

In[352]:= sol3 = integrate[eq2 + eps * vec1, 0.0, -0.005, 2000];

In[353]:= man3 = phaser[sol3];

In[354]:= sol4 = integrate[eq2 - eps * vec1, 0.0, -0.005, 2000];

In[355]:= man4 = phaser[sol4];

In[356]:= display = True;
```

In[357]:= `mangraph = show[man1, man2, man3, man4];`



Now we create a second piece of the portrait with some initial conditions near the saddlepoint.

In[358]:= `initlist = {{2, 1.5}, {0.75, 2}, {1.5, 0}, {0, 0.5}, {0, 3.2}, {3, 2}, {2.0, -1.5}};`

We show the curves in blue.

In[359]:= `setcolor[{Blue}];`

We integrate in both directions, with range checking.

In[360]:= `bothdirflag = True;`

In[361]:= `rangeflag = True;`

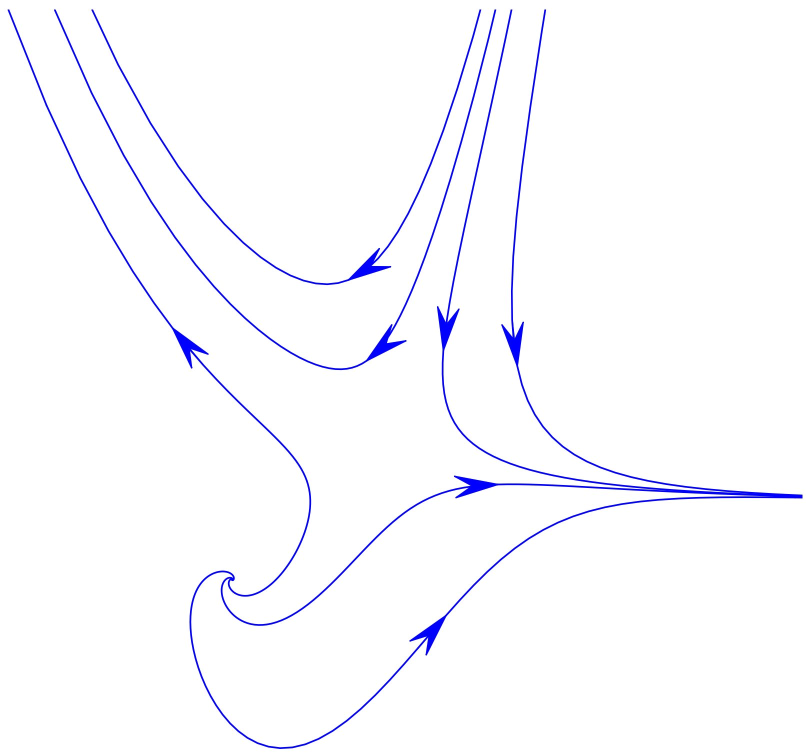
In[362]:= `ranger = {{-6, 8}, {-6, 8}};`

We ask for a suggested time step, and we will use half the suggested value.

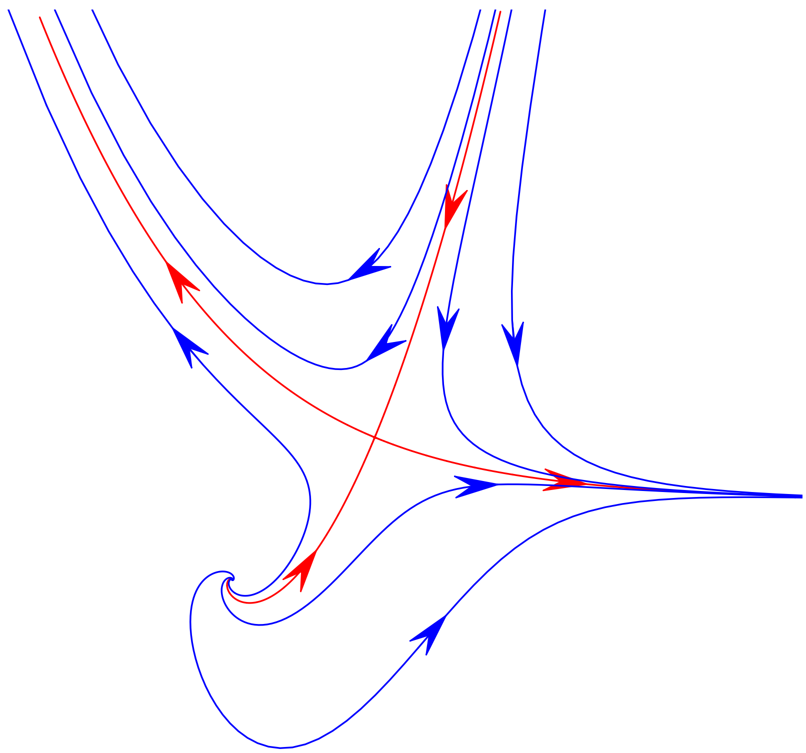
In[363]:= `h = sugtimestep[plrange]`

Out[363]= 0.111098

```
In[364]:= orbgraph1 = portrait[initlist, 0.0, h / 2, 600, 1, 2];
```



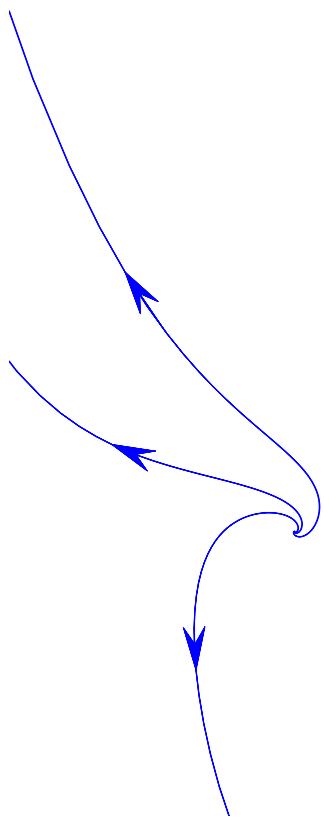
```
In[365]:= show[mangraph, orbgraph1];
```



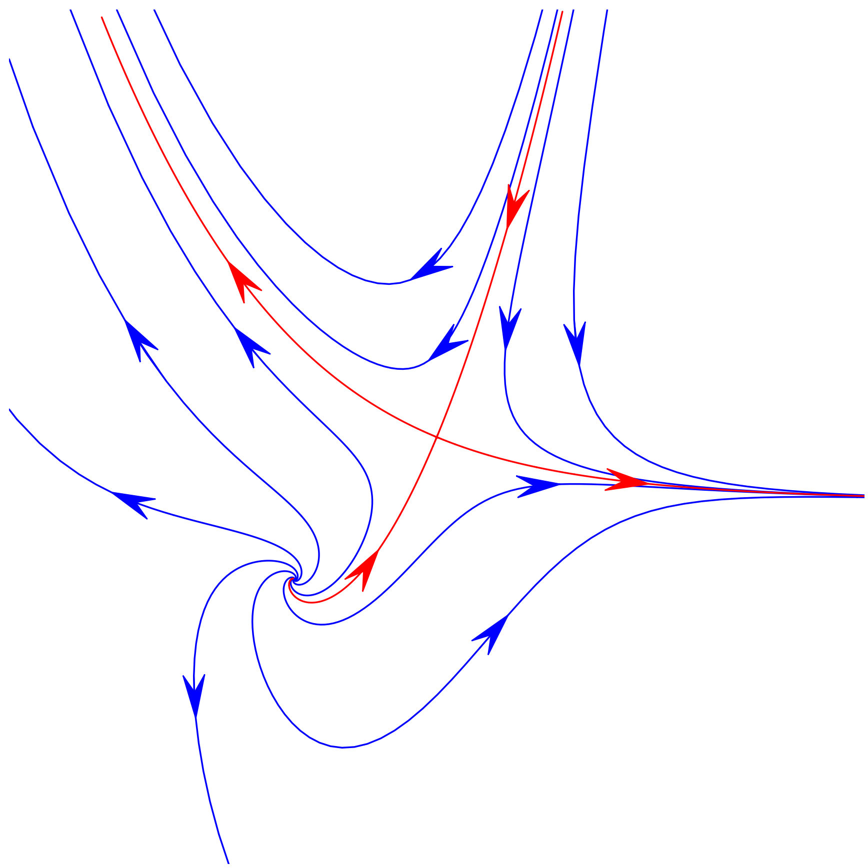
We add just a few more curves coming from the spiral.

```
In[366]:= initset = {{-1, 0}, {-2, -1}, {-3, 0}};
```

```
In[367]:= orbgraph2 = portrait[initset, 0.0, h / 2, 600, 1, 2];
```



```
In[368]:= fingraph = show[orbgraph1, orbgraph2, mangraph];
```



This is the graph that we will use to make the movie. The movie is made by constructing a sequence of graphs with an increasing plot range in the sequence. The sequence is constructed by the routine zoom[graph,smallwin,bigwin,steps] defined below. The arguments are graph, the name of the graph, smallwin, the smallest plotting window in the sequence, bigwin, the largest plotting window in the sequence, and steps, one less than the number of graphs in the sequence.

```
In[369]:= zoom[graph_, smallwin_, bigwin_, steps_] := Module[{i, tempplrange},
                tempplrange = plrange;
           Do[plrange = ((steps - i) / steps) * smallwin + (i / steps) * bigwin;
            show[graph], {i, 0, steps}]; plrange = tempplrange]
```

We try this with our present graph, with an initial window

```
In[370]:= smallwin = {{0.9, 1.1}, {0.9, 1.1}};
```

an a final window of

```
In[371]:= bigwin = {{-5, 7}, {-5, 7}};
```

We now create a 51-graph sequence. At the beginning of the sequence our view is so close to the saddle point that the stable and unstable manifolds appear to be straight lines, which is the local approximation. As we rise above the phase plane, we first see some curvature, and then later we see the spiral and its connection with the saddle point. In the printed version of this notebook, only the first frame of the movie is shown. To see the movie, double click on the graph below.

```
In[372]:= zoom[fingraph, smallwin, bigwin, 50];
```