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

```
sysid
```

```
Mathematica  6.0.3, DynPac 11.01, 1/13/2009
```

```
intreset;
```

```
plotreset; imsize = 250;
```

▪ 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.

```
setstate[{x, y}];
```

```
setparm[{}];
```

```
parmval = {};
```

```
slopevec = {x - y, 1 - x * y};
```

```
sysname = "";
```

▪ EQUILIBRIUM POINTS

We use findpolyeq to find the equilibrium points:

```
eqpoints = findpolyeq
```

```
{{-1, -1}, {1, 1}}
```

```
eq1 = eqpoints[[1]]
```

```
{-1, -1}
```

```
eq2 = eqpoints[[2]]
```

```
{1, 1}
```

We now classify these:

```
classify2D[eq1]
```

```
unstable - spiral
```

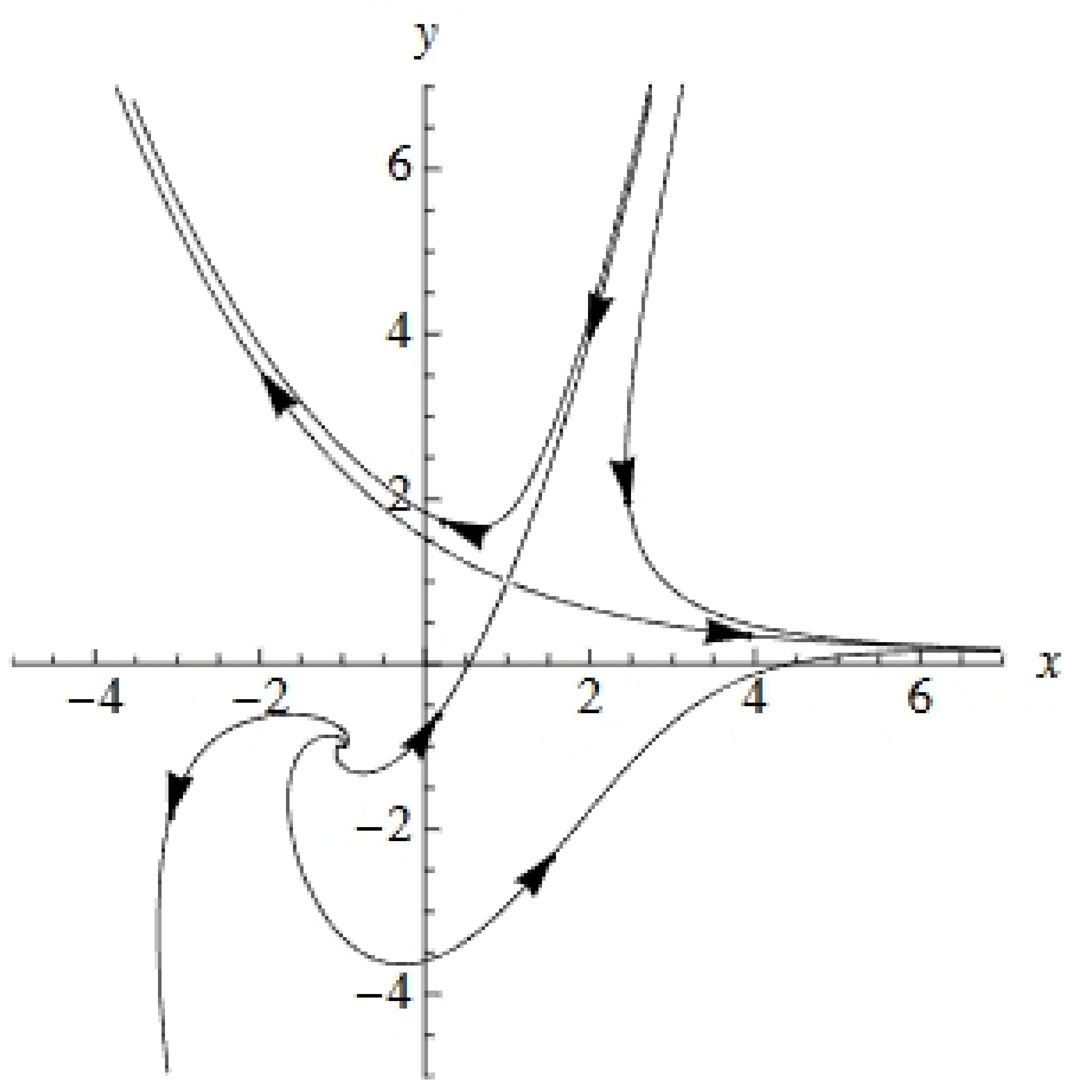
```
classify2D[eq2]
```

```
unstable - saddle
```

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

```
plrange = {{-5, 7}, {-5, 7}};
```

```
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.

```
eiger = Eigensystem[dermatval[N[eq2]]]
```

```
{{-1.41421, 1.41421}, {{0.382683, 0.92388}, {0.92388, -0.382683}}}
```

```
vec1 = First[Last[eiger]]
```

```
{0.382683, 0.92388}
```

```
vec2 = Last[Last[eiger]]
```

```
{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.

```
eps = 0.001;

rangeflag = True;

ranger = plrange;

arrowflag = True;

arrowvec = {1 / 2};

setcolor[{Red}];

axon = False;

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

man1 = phaser[sol1]
```

```
sol2 = integrate[eq2 - eps * vec2, 0.0, 0.005, 2000];
```