Homogeneous dynamics: Can be bi-stable

In[1]:=
```
deq = {u'[t] == u[t] - u[t]^3 - v[t],
  v'[t] == epsilon * (u[t] - a1 v[t] - a0),
  u[0] == u0, v[0] == v0}
```

Out[1]=
$$\{u'[t] == u[t] - u[t]^3 - v[t],\ v'[t] == \text{epsilon}\,(-a0 + u[t] - a1\,v[t]),\ u[0] == u0,\ v[0] == v0\}$$

In[2]:=
```
para = {a0 → -0.1, a1 → 2, epsilon → 0.05}
```

```
deqN1 = deq /. para /. {u0 → 0, v0 → 0}
```

Out[2]=
$$\{a0 \to -0.1,\ a1 \to 2,\ \text{epsilon} \to 0.05\}$$

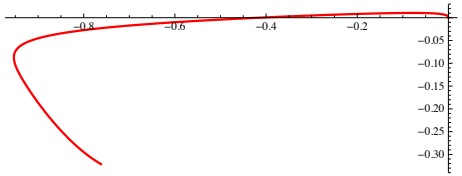
Out[3]=
$$\{u'[t] == u[t] - u[t]^3 - v[t],\ v'[t] == 0.05\,(0.1 + u[t] - 2\,v[t]),\ u[0] == 0,\ v[0] == 0\}$$

In[4]:=
```
time = 30;
sol1 = NDSolve[deqN1, {u[t], v[t]}, {t, 0, time}]
```

Out[5]=
$$\{\{u[t] \to \text{InterpolatingFunction}[\{\{0., 30.\}\}, <>][t],\ v[t] \to \text{InterpolatingFunction}[\{\{0., 30.\}\}, <>][t]\}\}$$

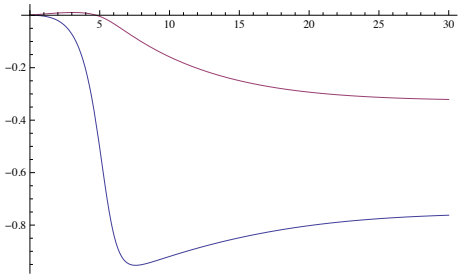
In[6]:=
```
g1 = ParametricPlot[ Evaluate[{u[t], v[t]} /. sol1], {t, 0, time},
  PlotRange → All, PlotStyle → {Thickness[0.005], RGBColor[1, 0, 0]}]
```

Out[6]=



In[7]:=
```
Plot[ Evaluate[{u[t], v[t]} /. sol1], {t, 0, time}, PlotRange → All]
```

Out[7]=



In[8]:=
```
deqN2 = deq /. para /. {u0 → -1, v0 → -0.8}
```

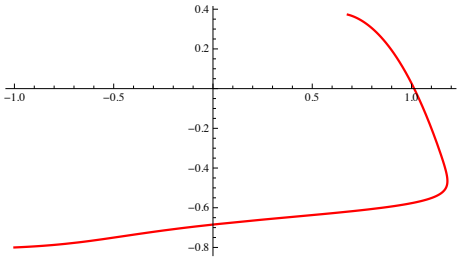
Out[8]=
$$\{u'[t] == u[t] - u[t]^3 - v[t],\ v'[t] == 0.05\,(0.1 + u[t] - 2\,v[t]),\ u[0] == -1,\ v[0] == -0.8\}$$

In[9]:=
```
time = 30;
sol2 = NDSolve[deqN2, {u[t], v[t]}, {t, 0, time}]
```

Out[10]=
$$\{\{u[t] \to \text{InterpolatingFunction}[\{\{0., 30.\}\}, <>][t],\ v[t] \to \text{InterpolatingFunction}[\{\{0., 30.\}\}, <>][t]\}\}$$

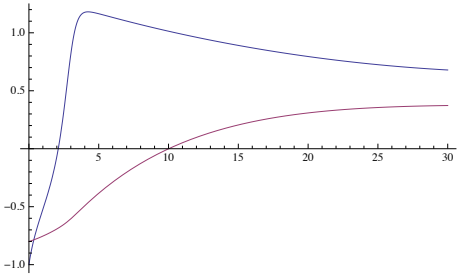
In[11]:=
```
g2 = ParametricPlot[ Evaluate[{u[t], v[t]} /. sol2], {t, 0, time},
  PlotRange → All, PlotStyle → {Thickness[0.005], RGBColor[1, 0, 0]}]
```

Out[11]=



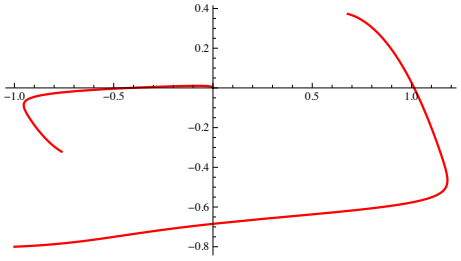
In[12]:=
```
Plot[ Evaluate[{u[t], v[t]} /. sol2], {t, 0, time}, PlotRange → All]
```

Out[12]=



In[13]:=
```
Show[g1, g2]
```

Out[13]=



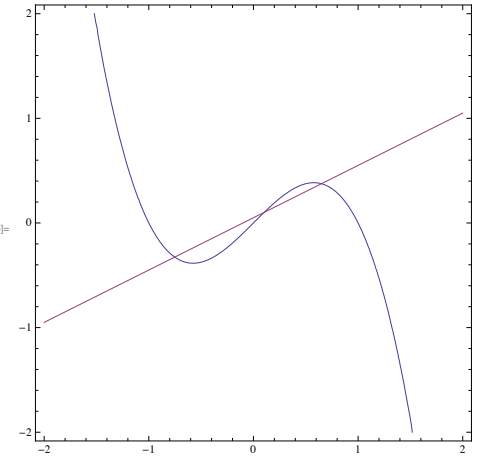
In[14]:=
```
fn1 = u - u^3 - v /. para
fn2 = epsilon (u - a1 v - a0) /. para
```

Out[14]=
$$u - u^3 - v$$

Out[15]=
$$0.05\,(0.1 + u - 2\,v)$$

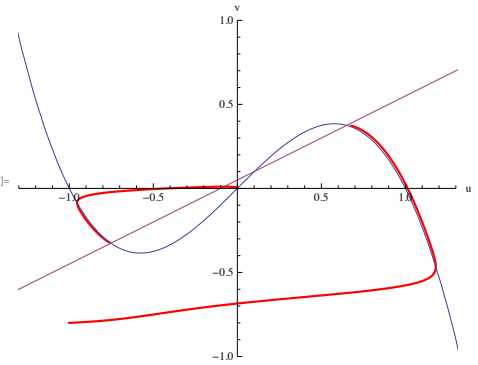
In[16]:=
```
gIso = ContourPlot[{fn1 == 0, fn2 == 0}, {u, -2, 2}, {v, -2, 2}]
```

Out[16]=



In[17]:=
```
Show[g1, g2, gIso,
 PlotRange → {{-1.3, 1.3}, {-1, 1}}, AxesLabel → {"u", "v"}]
```

Out[17]=



In[18]:=
```
FindRoot[{fn1 == 0, fn2 == 0}, {u, 0.8}, {v, 1}]
```

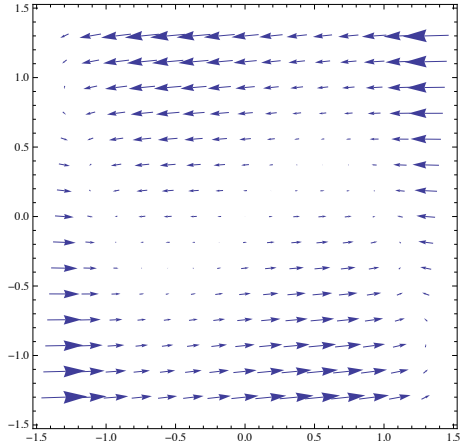
Out[18]=
```
{u → 0.650488, v → 0.375244}
```

In[19]:=
```
FindRoot[{fn1 == 0, fn2 == 0}, {u, -0.8}, {v, -1}]
```

Out[19]=
```
{u → -0.752619, v → -0.326309}
```

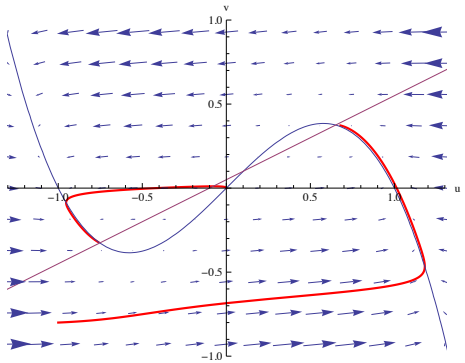

In[20]:=
```
gVec = VectorPlot[{fn1, fn2}, {u, -1.3, 1.3},
  {v, -1.3, 1.3}, VectorScale → {0.09, Automatic, Automatic}]
```

Out[20]=



In[21]:=
```
Show[g1, g2, gIso, gVec,
 PlotRange → {{-1.3, 1.3}, {-1, 1}}, AxesLabel → {"u", "v"}]
```

Out[21]=



In[22]:=
```
? VectorPlot
```

VectorPlot[{v_x, v_y}, {x, x_{min}, x_{max}}, {y, y_{min}, y_{max}}]
　　ベクトル場{v_x, v_y}のベクトルプロットを x と y の関数として生成する.
VectorPlot[{{v_x, v_y}, {w_x, w_y}, …}, {x, x_{min}, x_{max}}, {y, y_{min}, y_{max}}] 複数のベクトル場をプロットする.　≫

2

2