Bifurcation Diagrams - 1D ODEs

Nonlinear Systems and Modeling

Katie Oliveras

Preliminaries

Let's assume that you want to examine the bifurcation diagram associated with the differential equation $\dot{x} = f(x; a)$ where a is a real-valued parameter/constant. For this example, let's consider the differential equation where we have $\dot{x} = x^3 - a\,x$. First, we want to define the function $f(x; a) = x^3 - a\,x$ so that we can examine the equilibrium points.

In[17]:=
```
f[x_, a_] = x^3 - a x
```

Out[17]= $-a\,x + x^3$

Next, let's solve for the equilibrium points by having Mathematica solve the equation $\dot{x} = 0 \;\; \rightarrow \;\; f(x; a) = 0$. The command below solves the equation $f(x; a) = 0$ for x and returns a list of solutions.

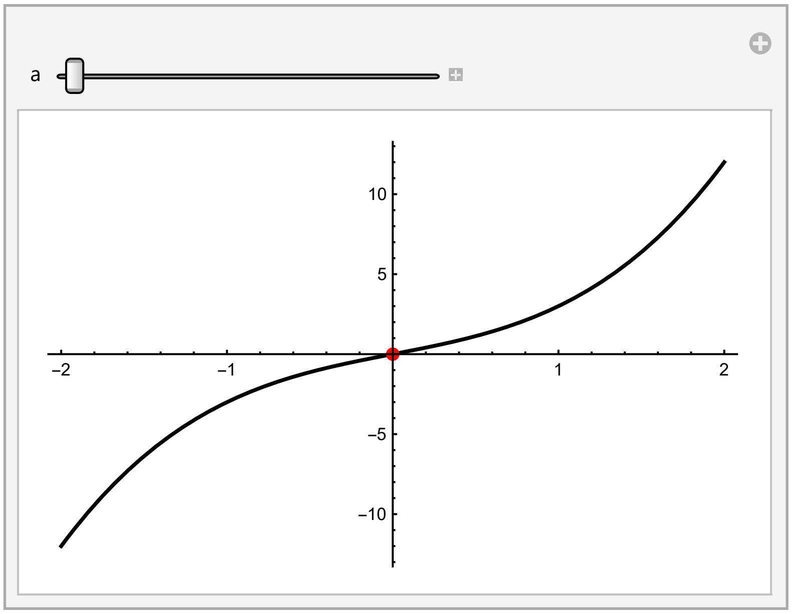
In[18]:=
```
eqPts[a_] = Solve[f[x, a] == 0, x]
```

Out[18]= $\{\{x \to 0\}, \{x \to -\sqrt{a}\}, \{x \to \sqrt{a}\}\}$

Now that we know the equilibrium points, we would like to investigate the phase line and solution curves as a function of the parameter a. The following in a simple way to plot the functions for yourself and see how the phase line changes using the MANIPULATE command

In[19]:=
```
Manipulate[
 Show[
  Plot[f[x, a], {x, -2, 2}, PlotStyle → {Black, Thick}],
  ListPlot[{x, 0} /. eqPts[a], PlotStyle → {Red, PointSize[Large]}]],
 {a, -2, 2}]
```

Out[19]=



General Code To Display Side-by-Side-by-Side

The following code allows your to sketch the Phase Lines, the Bifurcation Diagram, and the Solution Curves for a differential equation of the form $\dot{x} = f\,(x; a)$ where you have created the functions $f\,[x\_, a\_]$ and eqPts $[a\_]$

In[29]:=

```
PhaseLinePlot[a_] := Module[{eqPoints = eqPts[a], p1, p2},
  p1 = Plot[f[x, a], {x, -2, 2}, ImageSize → 300,
    PlotRange → {-1.5, 1.5}, PlotStyle → {Black, Thick}];
  p2 = ListPlot[Table[{x /. eqPoints⟦j⟧, 0}, {j, 1, Length[eqPoints]}],
    PlotStyle → {Red, PointSize[Large]}];
  Show[p1, p2]
 ]
BifurcationPlot[α_] := Module[{ plotStable, plotUnstable, eqPtPlot, condition},
  For[j = 1, j ≤ Length[eqPts[a]], j++,
   condition_j = (D[f[x, a], x] /. eqPts[a]⟦j⟧) > 0;
   plotStable_j =
    Plot[If[condition_j, x /. eqPts[a]⟦j⟧], {a, -1, 1}, PlotStyle → {Black, Dashed, Thick}];
   plotUnstable_j = Plot[If[Not[condition_j], x /. eqPts[a]⟦j⟧],
     {a, -1, 1}, PlotStyle → {Black, Thick}]];
  eqPtPlot = ListPlot[Table[{α, x /. eqPts[α]⟦j⟧}, {j, 1, Length[eqPts[α]]}],
    PlotStyle → {Red, PointSize[Large]}];
  Show[Flatten[{Table[{plotStable_j, plotUnstable_j}, {j, 1, Length[eqPts[a]]}], eqPtPlot}],
   PlotRange → All, ImageSize → 300]
 ]
SolutionPlot[α_] := Module[{p1, p2},
  p1 = StreamPlot[{t, f[x, α]}, {t, 0, 10}, {x, -1.5, 1.5}, ImageSize → 300];
  p2 = Plot[Table[x /. eqPts[α]⟦j⟧, {j, 1, Length[eqPts[α]]}],
    {t, 0, 10}, PlotStyle → {Black, Dashed, Thick}];
  Show[p1, p2]]

allPlots[α_] := Row[{PhaseLinePlot[α], BifurcationPlot[α], SolutionPlot[α]}]
```

To call the code, simply use the `Manipulate` command as shown below:

In[24]:=
```
Manipulate[allPlots[α], {α, -1, 1}]
```

Out[24]=