Lecture 25: Phase Plane Analysis and Critical Points

Reading:
Kreyszig Sections: 4.1, 4.2 (pages131–135, 136–139)

Phase Plane and Critical Points

A few examples of physical models that can be represented by systems of first-order differential equations:

$$\frac{d\vec{y}}{dt} \equiv \frac{d}{dt}\begin{pmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_N(t) \end{pmatrix} = \begin{pmatrix} F_1(y_1, y_2, \ldots, t) \\ F_2(y_1, y_2, \ldots, t) \\ \vdots \\ F_N(y_1, y_2, \ldots, t) \end{pmatrix} = \begin{pmatrix} F_1(\vec{y}, t) \\ F_2(\vec{y}, t) \\ \vdots \\ F_N(\vec{y}, t) \end{pmatrix} \equiv \vec{F}(\vec{y}, t) \tag{25-1}$$

and, furthermore, it has been shown that many higher-order systems of ODEs can be reduced to larger systems of first-order ODEs.

The behavior of systems of first-order equations can be visually interpreted by plotting the trajectories $\vec{y}(t)$ for a variety of initial conditions $\vec{y}(t=0)$. An illustrative example is provided by the equation for the pendulum, $MR^2\ddot{\theta} + MgR\sin\theta = 0$. can be re-written with the angular momentum ω as the system of first-order ODEs

$$\begin{aligned} \frac{d\theta}{dt} &= \frac{\omega}{MR} \\ \frac{d\omega}{dt} &= -Mg\sin\theta \end{aligned} \tag{25-2}$$

which was shown in Lecture 22 to have solutions:

$$\frac{\omega^2}{2M} - MgR\cos\theta = E_o \tag{25-3}$$

Eq. 25-3 can be used to plot the the trajectories in the phase plane.

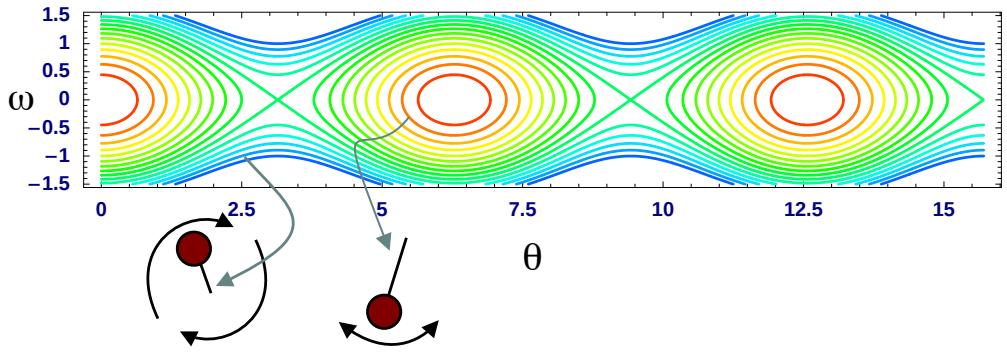

Figure 25-26: Example of the phase plane for the pendulum equation. The small closed orbits are the stable harmonic oscillations about the stable position $\downarrow$. The larger orbits are those with increasing energy until the energy is just large enough that the pendulum rises to its unstable equilibrium position $\uparrow$. The two kinds of fixed points (i.e., the stable and unstable points where $\dot{\omega} = \dot{\theta} = 0$) regulate the portrait of the phase plane. (Note: The word "phase" here should not be confused with the common usage of phase in materials science. In the current context for example, the phase represents the positions and momenta of all the particles in a system—this usage is important in statistical mechanics. However, the word "phase" in materials science and engineering is usually interpreted as a portion of material that lies within an identifiable interface—this usage is implied in "equilibrium phase diagrams.")

Behavior for a wide variety of initial conditions can be comprehended by the following approach:

Identify Fixed Points If all the points in the phase plane where $d\vec{y}/dt = 0$ can be established, then these fixed points can be used as reference points around which the phase-behavior will be determined.

Linearization At each fixed point, Linearization is obtained by expanding Eq. 25-1 to first order in $\vec{\eta} = \vec{y} - \vec{y}_{\text{fixed}}$, the zeroth-order term vanishes by construction:

$$\frac{d}{dt}\begin{pmatrix} \eta_1(t) \\ \eta_2(t) \\ \vdots \\ \eta_N(t) \end{pmatrix} = \begin{pmatrix} \frac{\partial F_1}{\partial y_1}\Big|_{\vec{y}_{\text{fixed}}} & \frac{\partial F_1}{\partial y_2}\Big|_{\vec{y}_{\text{fixed}}} & \cdots & \frac{\partial F_1}{\partial y_N}\Big|_{\vec{y}_{\text{fixed}}} \\ \frac{\partial F_2}{\partial y_1}\Big|_{\vec{y}_{\text{fixed}}} & \ddots & & \\ \vdots & & & \vdots \\ \frac{\partial F_N}{\partial y_1}\Big|_{\vec{y}_{\text{fixed}}} & \cdots & \cdots & \frac{\partial F_N}{\partial y_N}\Big|_{\vec{y}_{\text{fixed}}} \end{pmatrix} \begin{pmatrix} \eta_1 \\ \eta_2 \\ \vdots \\ \eta_N \end{pmatrix} \tag{25-4}$$

Eigenvalues/Eigenvectors When the system Eq. 25-4 is transformed into a coordinate frame in which the matrix is diagonal, then each component of $\vec{\eta}_{\text{eigen-frame}}$ has a trajectory that is unaffected by the others and determined by only the diagonal entry associated with that component.

The $\vec{\eta}_{\text{eigen-frame}}$ are the eigenvectors of Eq. 25-4 and the diagonal component is its associated eigenvalue.

Fixed Point Characterization If the eigenvalue is real, then any point that lies in the direction of its eigenvector will evolve along a straight path parallel to the eigenvector. If the real eigenvalue is negative, that straight path will asymptotically approach the origin; if the eigenvalue is positive the trajectory will diverge along the straight-path towards infinity.

If the eigenvalue is imaginary, then the trajectory will circulate about the fixed point with a frequency proportional the eigenvalue's magnitude.

If the eigenvalue, λ is complex, its trajectory will both circulate with a frequency proportional to its imaginary part and diverge from or converge to the fixed point according to $\eta_o \exp(\mathrm{Re}\lambda)$.

If any one of the fixed points has an eigenvalue with a positive real part, the fixed point cannot be stable—this is because "typical" points in the neighborhood of the fixed points will possess some component of the unstable eigenvector.

Stability of Critical Points

For the two-dimensional linear system

$$\frac{d}{dt}\begin{pmatrix} \eta_1(t) \\ \eta_2(t) \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} \eta_1 \\ \eta_2 \end{pmatrix} \tag{25-5}$$

can be analyzed because the eigenvalues can be calculated directly from the quadratic equation.

Every two-by-two matrix has two invariants (i.e., values that do not depend on a unitary transformation of coordinates). These invariants are the *trace,* T of the matrix (the sum of all the diagonals) and the determinant D. The eigenvalue equation can be written in terms of these two invariants:

$$\lambda^2 - T\lambda + D = 0 \tag{25-6}$$

The *discriminant* $\Delta \equiv T^2 - 4D$ appears in the solutions to the eigenvalues:

$$\lambda_\pm = \frac{T \pm \sqrt{\Delta}}{2} \tag{25-7}$$

There are five regions of behavior:

$\Delta \geq 0$ The eigenvalues are real.

Eigenvalues both positive ***An Unstable Node***: All trajectories in the neighborhood of the fixed point will be directed outwards and away from the fixed point.

Eigenvalues both negative ***A Stable Node***: All trajectories in the neighborhood of the fixed point will be directed towards the fixed point.

Eigenvalues opposite sign ***An Unstable Saddle Node***: Trajectories in the general direction of the negative eigenvalue's eigenvector will initially approach the fixed point but will diverge as they approach a region dominated by the positive (unstable) eigenvalue.

$\Delta < 0$ Eigenvalues are complex conjugates—their real parts are equal and their imaginary parts have equal magnitudes but opposite sign.

Real parts positive ***An Unstable Spiral***: All trajectories in the neighborhood of the fixed point spiral away from the fixed point with ever increasing radius.

Real parts negative ***An Stable Spiral***: All trajectories in the neighborhood of the fixed point spiral into the fixed point with ever decreasing radius.

The curves separating these regions have singular behavior. For example, where $T = 0$ for positive D, the eigenvalues are purely imaginary and trajectories circulate about the fixed point in a stable orbit. This is called a ***center*** and is the case for an undamped harmonic oscillator.

The regions can be mapped with the invariants and the following diagram illustrates the behavior.

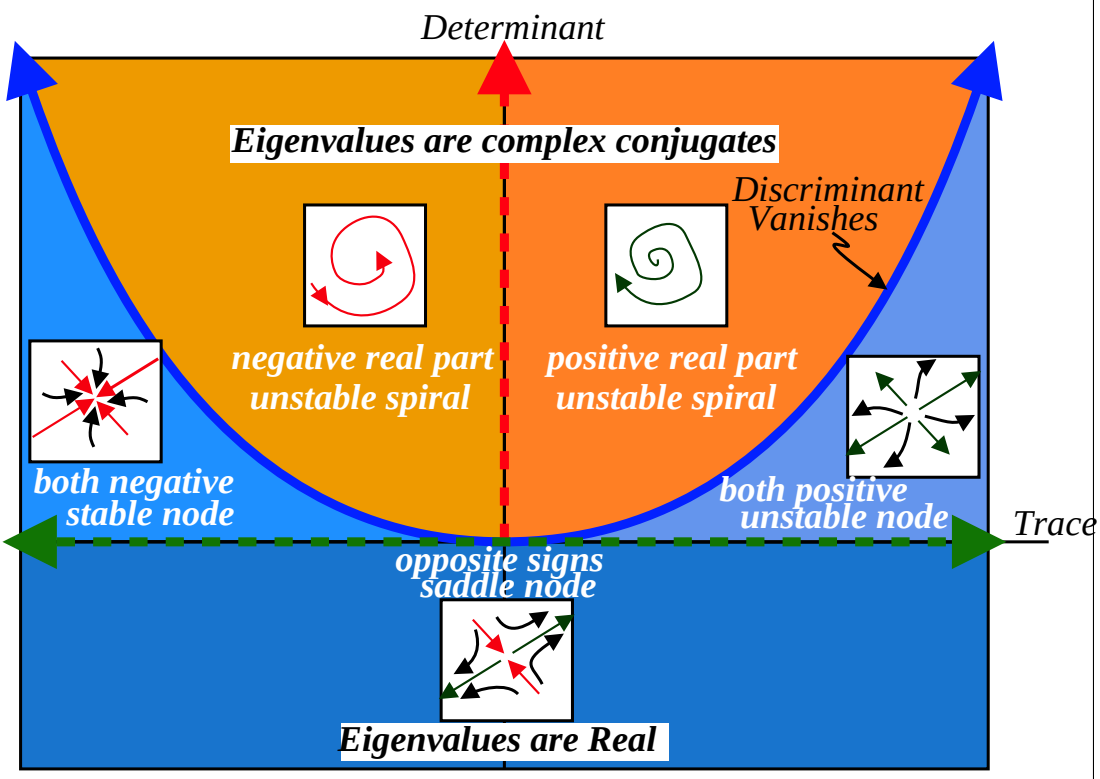

Figure 25-27: Illustration of the five regions according to their behavior near the fixed point.

At the point where the five regions come together, all the entries of the matrix of coefficients are zero and the physical behavior is then determined by expanding Eq. 25-1 to the next highest order at which the coefficients are not all zero.

Lecture 25 MATHEMATICA® Example 1

Functions to Analyze Fixed Points for Two-Dimensional Systems

Download notebooks, pdfs, or html from http://pruffle.mit.edu/3.016-2006.

Several functions are defined that identify the type of fixed point, their stability and the orientation of the stable and unstable eigenvectors.

1: *Eigenconsistency* takes two eigenvalues as arguments and checks if they derive from a real 2×2 matrix. If the eigenvalues are complex and the matrix is real, then the two eigenvalues must be complex conjugates.

2: *Eigenstability* determines the sign of the real part of each eigenvalue and uses `Print` to display a friendly message about the solution's stability.

3: From the nature of the two eigenvalues, *EigenTrajectory* will print a friendly message about the *fixed point type*.

4: *EigenDescription* collects the previous three messaging functions into a single function.

6: *EigenDirector* takes a the stucture resulting from `Eigensystem` and uses the orientation of the eigenvectors and `ArcTan` to describe the local orientation of any stable or unstable directions.

7: *LinearDescription* takes the entries of a 2×2 matrix, calculates the eigensystem, removes any numerically trivial parts with `Chop` and then proceeds to call all the previous messaging functions to give a complete description of the fixed points behavior.

```
1  EigenConsistency[eval1_, eval2_] :=
     If[And[And[Im[eval1] ≠ 0, Im[eval2] ≠ 0],
         Conjugate[eval1] ≠ eval2],
       Print["Coefficients are not real"]]

2  EigenStability[eval1_, eval2_] :=
     If[And[Re[eval1] < 0, Re[eval2] < 0],
       Print["Stable and Attractive"],
       If[Or[Re[eval1] > 0, Re[eval2] > 0], Print["Unstable"],
         Print["Stable Orbits about Fixed Point"]]]

3  EigenTrajectory[eval1_, eval2_] :=
     If[Or[Im[eval1] ≠ 0 , Im[eval2] ≠ 0], Print["Circulation"],
       If[(s1 = Sign[Re[eval1]]) ≠ (s2 = Sign[Re[eval2]]),
         Print["Saddle"], Print["Node"]]]

4  EigenDescription[eval1_, eval2_] :=
     Module[{}, EigenConsistency[eval1, eval2];
       EigenStability[eval1, eval2]; EigenTrajectory[eval1, eval2]]

5  EigenDescription[ −1 + i, −1 − i]
   EigenConsistency[1 + i, 1]

6  EigenDirector[eval_, {ex_, ey_}] :=
     Module[{theta = N[180 * ArcTan[ex, ey] / π]}, If[eval > 0,
         Print["Unstable (λ=", eval, ") direction is θ = ", theta]];
       If[eval < 0, Print["Stable (λ=", eval,
           ") direction is θ = ", theta]]]

7  LinearDescription[a_, b_, c_, d_] :=
     Module[{esys, eval1, eval2, evec1, evec2},
       esys = Eigensystem[{{a, b}, {c, d}}];
       eval1 = Chop[esys[[1, 1]]]; eval2 = Chop[esys[[1, 2]]];
       evec1 = Chop[esys[[2, 1]]]; evec2 = Chop[esys[[2, 2]]];
       EigenDescription[eval1, eval2];
       Print["Eigenvalues = ", eval1, " and ", eval2];
       If[And[Im[eval1] == 0, Im[eval1] == 0],
         EigenDirector[eval1, evec1];
         EigenDirector[eval2, evec2]]; esys]

8  LinearDescription[1, .1, 1, −3]
```

Lecture 25 MATHEMATICA® Example 2

Visualizing the Behavior at a Fixed Point in the Plane

Download notebooks, pdfs, or html from http://pruffle.mit.edu/3.016-2006.

A function for visualizing the behavior of a fixed point with oriented arrows is constructed.

```
LinSol[a_, b_, c_, d_] :=
  {x[t], p[t]} /. Flatten[
1       DSolve[{x'[t] == a x[t] + b p[t], p'[t] == c x[t] + d p[t],
          x[0] == Random[Real, {-10, 10}],
          p[0] == Random[Real, {-10, 10}]}, {x[t], p[t]}, t]]
```

```
CritPointPlot[a_, b_, c_, d_] :=
2  ParametricPlot[Evaluate[LinSol[a, b, c, d]],
    {t, 0, 20}, PlotRange → {{-15, 15}, {-15, 15}}]
```

```
3 << Graphics`Arrow`
```

```
CritPointPlotPointsMany[a_, b_, c_, d_, howmany_:100] :=
  Module[{esys, eval1, eval2, funcs , data, lendata, gstack, lp,
      rstack, lstack, magrange, rstackmag, lstackmag},
    (*A*) esys = LinearDescription[a, b, c, d]; Print[esys];
    eval1 = esys[[1, 1]]; eval2 = esys[[1, 2]];
    (*B*) funcs = Table[Chop[LinSol[a, b, c, d]],
        {i, howmany}]; (*C*) lstack = {}; rstack = {};
    For[imany = 1, imany ≤ howmany, imany++,
      (*D*)data = Chop[Table[Evaluate[
            funcs[[imany]] /. t → itime], {itime, 0, 20, .1}]];
      lendata = Length[data]; gstack = {Hue[0], Arrow[
          data[[1]], data[[2]], HeadScaling → Absolute]};
      (*E*) For[iend = 4, iend ≤ lendata/2, iend += 8,
        (*F*)AppendTo[gstack, Hue[iend * 0.66 * 2/lendata]];
        (*G*)AppendTo[gstack, Arrow[data[[iend - 1]],
4         data[[iend]], HeadScaling → Absolute]]];
      (*H*)lp = ListPlot[data, PlotJoined → True,
        AspectRatio → 1, PlotRange → {{-15, 15}, {-15, 15}},
        PlotStyle → Hue[Random[]],
        DisplayFunction → Identity];
      (*I*)lstack = Show[lp, Graphics[gstack], lstack];
      (*J*)rstack = Show[lp, rstack]];
    (*K*)If[Or[Re[eval1] > 0, Re[eval2] > 0],
      magrange = {{-60, 60}, {-60, 60}},
      magrange = {{-1, 1}, {-1, 1}}];
    rstackmag = Show[rstack, PlotRange → magrange];
    lstackmag = Show[lstack, PlotRange → magrange];
    (*L*)Show[
      GraphicsArray[{{lstack, rstack}, {lstackmag, rstackmag}}],
      ImageSize → 1000,
      DisplayFunction → $DisplayFunction]]
```

```
5 CritPointPlotPointsMany[-1, 0.25, 0.75, -1]
```

1: *Linsol* , with four arguments representing the Jacobian entries, calculates the solution for an initial point picked randomly from the domain $(-10 \leq x \leq 10), (-10 \leq y \leq 10)$. In this case, the fixed point is assumed to have been translated to the origin. It calls `DSolve`, on a coupled pair of first-order ODEs and uses `Random` to generate the initial conditions.

2: This function takes the entries for the Jacobian and then uses `ParametricPlot` to plot the solution returns from *LinSolve* . However, the results are not very easy to interpret: one can't discern the direction of the tragjectory, nor the nature of an fixed point.

3: `Graphics‘Arrow‘` has arrow graphics objects to indicate the direction.

4: *CritPointPlotPointsMany* is an elaborate function that provides four graphical views of a critical point by computing and plotting a large number of trajectories emanating from random intial points. This function illustrates an example of the use of *optional function-arguments* in `howmany_:100` (i.e., the number of random trajectories to compute) which defaults to 100 if that argument is left off the function when it is called.

4A Compute the eigensystem and assign it to `esys`, *LinearDescription* will also print information about the critial point.

4B `funcs` is a list of solution-pairs for different initial points.

4C `Show` will be called recursively to add graphical objects to a list `lstack`. `lstack` is a placeholder for plots that contain arrows. Items **D**–**J** are in the body of a loop over each of the `homany` trajectories. **4D**: data is created for 20 discrete points at equal $\Delta t = 0.1$ along the current trajectory. A red arrow object is created at the beginning of the trajectory and stored in `gstack`. **4E**: This will loop over points the first half of the data. Items **F**–**G** are within the loop's body. **4F**: The color of the arrow will shift towards blue with time. **4G**: The arrow object is added to `gstack`. **4H**: The data is plotted with `ListPlot`. **4I**: This combines the contents of the plot and the two graphics-lists and assigns it to `lstack`—this iteratively grows `lstack`. **4J**: `rstack` will only contain plots and no arrows. **4K**: To focus on the long-time behavior, the `PlotRange` will need to be large (like a telescope) for unstable systems and small (like a microscope) for stable fixed points. The `If` will pick the appropriate `PlotRange`. **4L**: `GraphicsArray` produces an array of plots.

Unstable Manifolds

The phase portraits that were visualized in the above example help illustrate a very powerful mathematical method from non-linear mechanics.

Consider the saddle-node that has one positive (unstable) and one negative (stable) eigenvalue. Those initial points that are located in regions where the negative (stable) eigenvalue dominates are quickly swept towards the fixed point and then follow the unstable direction away from the fixed point. Roughly speaking, the stable values are 'smashed' onto the unstable direction and virtually all of the motion takes place near the unstable direction.

This idea allows a large system (i.e., one in which the vector $\vec{y}(t)$ has many components) to be reduced to a smaller system in which the stable directions have been approximated by a thin region near the trajectories associated with the unstable eigenvalues. This is sometimes called reduction of "fast variables" onto the unstable manifolds.