Phase Portraits of Nonlinear Differential Equations

Nonlinear Differential Equations:

Consider the system
\[
\begin{align*}
    x' &= f(x, y) \quad (1) \\
    y' &= g(x, y) \quad (2)
\end{align*}
\]
where \( f \) and \( g \) are functions of two variables \( x \) and \( y \).

The critical points (equilibrium points, steady state solutions) are found by setting \( x' = y' = 0 \), and solving the resulting system \( f(x, y) = 0 \) and \( g(x, y) = 0 \).

We approximate the nonlinear system by a linear system near the equilibrium points.
Suppose that \( (a, b) \) is an equilibrium point, that is \( f(a, b) = g(a, b) = 0 \). The linearized system is:
\[
\begin{align*}
    x' &= \partial_x f(a, b) x + \partial_y f(a, b) y \quad (1) \\
    y' &= \partial_x g(a, b) x + \partial_y g(a, b) y \quad (2)
\end{align*}
\]
whose coefficient matrix is the Jacobian matrix
\[
J(a, b) = \begin{pmatrix}
    \partial_x f(a, b) & \partial_y f(a, b) \\
    \partial_x g(a, b) & \partial_y g(a, b)
\end{pmatrix}.
\]

We can use the eigenvalues of the matrix Jacobian matrix to decide the type and stability of the critical point \( (a,b) \) (sink, source, saddle, etc.).

Jacobian Matrix

The following code from last class helps compute the Jacobian matrix of a function \( f \) of two variables \( x \) and \( y \):

```mathematica
In[1]:= Clear[f, J, x, y]
(* Jacobian *)
J[f_]:= Module[{A = Table[0, 2, 2]},
  A[[1, 1]] = D[f[x, y]][[1]], (x)];
  A[[1, 2]] = D[f[x, y]][[1, y]]; 
  A[[2, 1]] = D[f[x, y]][[2]]; (x));
  A[[2, 2]] = D[f[x, y]][[2, y]]; 
  Return[A]}
(* Example *)
J[f[x_, y_]] := (x (1 - y), y (x - 1))
J[f]
```

Example 2:

Consider the system
\[
\begin{align*}
    x' &= y - x(x^2 + y^2) \quad (1) \\
    y' &= -x - y(x^2 + y^2) \quad (2)
\end{align*}
\]
Study the nature of the critical point \( (x,y) = (0,0) \).

```mathematica
In[4]:= Clear[h]
(* Define the function h *)
h[x_, y_]:= (y - x (x^2 + y^2), -x - y (x^2 + y^2))
(* Find the critical points of the system *)
sol = Solve[y - x (x^2 + y^2) = 0, -x - y (x^2 + y^2) = 0], {x, y}]
Print[“Critical point: ”, sol]
(*Find the Jacobian matrix *)
Print[“The Jacobian Matrix is ” , MatrixForm[J[h]]]
(* Evaluate the Jacobian matrix at the only critical point (0,0) *)
A = J[h] /. Flatten[sol];
Print[“The Jacobian Matrix at (0,0) is ”, MatrixForm[A]]
Print[“Eigenvalues: “, Eigenvalues[A]]
```
Critical point: \((x=0, y=0)\)

The Jacobian Matrix is

\[
\begin{bmatrix}
-3x^2 - y^2 & 1 - 2xy \\
-1 - 2xy & -x^2 - 3y^2
\end{bmatrix}
\]

The Jacobian Matrix at \((0,0)\) is

\[
\begin{bmatrix}
0 & 1 \\
-1 & 0
\end{bmatrix}
\]

Eigenvalues: \(i, -i\) (complex conjugate with Real part equal to 0), so the origin is a center for the linearized system. Let’s see if \((0,0)\) is a center for the nonlinear system as well.

\[
\begin{align*}
\text{ODE5-net.nb} & \quad \text{ODE5-net.nb} \\
\text{ODE5-net.nb} & \quad \text{ODE5-net.nb}
\end{align*}
\]
Consider the system \[
\begin{aligned}
x' &= x+y-x(x^2+y^2) & (1) \\
y' &= -x+y-y(x^2+y^2) & (2)
\end{aligned}
\]
Find all critical points and study their nature.

Conclusions: The linearized system has a center at the origin. The nonlinear system does not have a center! The origin is a spiral sink.

Example 3

Conclusions: The origin is an spiral source (unstable spiral point) for both the
linear system and the nonlinear system. One may therefore think that all trajectories different from 0 should spiral out to infinity. This is clearly not the case, as far away from the origin, the arrows are pointing inward. Moreover, there is a closed trajectory (a circle) and all trajectories different from (0,0) spiral towards this circle (limit cycle). You can find the equation of this limit cycle by switching to polar coordinates \( x=r\cos\theta, y=r\sin\theta \).

### Polar Coordinates

I. We do the coordinate change:
\[
\begin{align*}
x &= r\cos\theta, \\
y &= r\sin\theta,
\end{align*}
\]
where \( r \) and \( \theta \) are functions of time \( t \).

II. We need to compute \( x' \) and \( y' \) as functions of \( r, \theta, r', \theta' \)

a) We can do it by hand, using the Chain Rule. We get:
\[
\begin{align*}
x' &= r'\cos\theta - r\sin\theta \frac{d\theta}{dt}, \\
y' &= r'\sin\theta + r\cos\theta \frac{d\theta}{dt}.
\end{align*}
\]

b) We use transformations rules in Mathematica to do the change of variables:
\[
\begin{align*}
x'[t] &= x'[r(t)] \cos[\theta(t)] + y'[r(t)] \sin[\theta(t)] \\
y'[t] &= x'[r(t)] \sin[\theta(t)] - y'[r(t)] \cos[\theta(t)]
\end{align*}
\]

Changing the system from Example 3 to polar coordinates

We know that \( x^2 + y^2 = r^2 \), so can use the symmetries of the system:
\[
\begin{align*}
x' &= x + y - x(x^2 + y^2), \\
y' &= y - x - y(x^2 + y^2).
\end{align*}
\]

III. Let \( s = x + y \) and \( y' - x' = y + x - y(x^2 + y^2) - x - x = 2y - x^2 = r^2 \)

\( s' = 0 \), therefore we can look at the following system:
\[
\begin{align*}
x' &= x + y - x(x^2 + y^2), \\
y' &= y - x - y(x^2 + y^2).
\end{align*}
\]

From the equation \( r' = r(1-r^2) \), we also see that:

- \( r' > 0 \) when \( r < 1 \) (thus the inside the unit circle the trajectories go outwards)
- \( r' < 0 \) when \( r > 1 \) (thus outside the unit circle the trajectories are directed inwards)

What about the other trajectories?
When \( r = 1 \) and \( r \neq 0 \), we can use separation of variables to solve the equation:
\[
\frac{dr}{dr(1-r^2)} = \frac{1}{r^2 - 1} \]

We write \( \frac{dr}{r^2 - 1} = dt \) and we can integrate each term separately.

\[
\int \frac{1}{r(1-r^2)} dr = \int \frac{1}{r-1} - \frac{1}{r+1} dr
\]

\[
\int \frac{1}{r(1-r^2)} dr = \log[r(1-r^2)] + \log[r-1] - \log[r+1]
\]

The equation \( \theta' = -1 \), solves to \( \theta(t) = -t + c \), where \( c \) is any real constant.

The critical points of \( r' = r(1-r^2) \) are:

- \( r=0 \) (the origin of the old phase portrait) and \( r=1 \) (the unit circle in the old phase portrait).

Therefore, a periodic solution of the system is \( r(t) = 1, \theta(t) = -t + c \).

(Assume \( t \) increases; a point moves clockwise on the unit circle)
The equation becomes \( \log(r) - \frac{1}{2} \log(1 - r^2) = t + k \), where \( k \) is any real constant, with solution given by \( r(t) = \frac{2^{2k+1}}{1 + 2^{2k+1}} \).

Nonlinear Equations with several variables:

In 1963, Lorenz studied a very simple model of the atmosphere. This model exhibits chaotic behavior:

Lorenz equation

\[
\begin{align*}
\dot{x} &= -10 (x(t) - y(t)), \\
\dot{y} &= -x(t) z(t) + 28 x(t) - y(t), \\
\dot{z} &= x(t) y(t) - \frac{8}{3} z(t), \\
\end{align*}
\]

(\( x, y, z \), (t, 0, tmax), MaxSteps \( \to \) 5000)
The 3D vector field can be plotted using the command `VectorPlot3D`

```math
VectorPlot3D[{-10 (x - y), -x + z + 28 x - y, x + y - 8 z},
  {x, -10, 10}, {y, -10, 10}, {z, -10, 10},
  VectorScale -> {Small, Automatic, None}]
```

The solution can be plotted using `ParametricPlot3D` as follows:

```math
Clear[x, y, z, t]
ParametricPlot3D[Evaluated[{x[t], y[t], z[t]} /. sol],
  {t, 0, tmax}, PlotPoints -> 3000, Axes -> False, PlotRange -> All]
```
Clear[{x, y, z, t}]
Manipulate[
    ParametricPlot3D[
        Evaluate[{x[t], y[t], z[t]} /. NDSolve[
            {x'[t] == -10 (x[t] - y[t]), y'[t] == -x[t] z[t] + 28 x[t] - y[t],
             z'[t] == x[t] y[t] - 0.3 z[t], x[0] == 30, y[0] == 10, z[0] == 40},
            {x, y, z}, {t, 0, T}, MaxSteps -> 5000],
            {t, 0, T}, PlotStyle -> Red],
            {{T, 2}, 0.1, 20}, SynchronousUpdating -> False,
            SaveDefinitions -> True]