Sukzessives Aufblasen einer Spitze

Ausgangssystem hat einen stationären Punkt um Ursprung. Jacobi-Matrix ist nilpotent.

```
f_0 := matrix(2,1,[ y, x^2+x*y]);
sol := solve(f_0,[x,y]); sol := sol[1]:
subs(jacobian(f_0,[x,y]),sol)
```

$$\begin{pmatrix} y \\ x^2 + y\,x \end{pmatrix}$$

$$\{[x = 0,\ y = 0]\}$$

$$\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$$

Bestimmung des minimalen Grades im Vektorfeld und des charakteristischen Polynoms.

```
m := min(op(map(f_0,ldegree)));
fm := map(map(map(f_0,monomials), select, p -> degree(p)=m), _plus@op);
F := factor(x*fm[2]-y*fm[1])
```

$$1$$

$$\begin{pmatrix} y \\ 0 \end{pmatrix}$$

$$-y^2$$

y=0 ist die einzige charakteristische Richtung. Nach einer y-Aufblasung finden wir keinen stationären Punkt mehr.

```
Psi_y := [ xx*yy, yy ];
SPsi_y := [ x=xx*yy, y=yy ]:
J_y := jacobian(Psi_y, [xx,yy]):
JI_y := simplify(J_y^(-1)):
```

$$[xx\ yy,\ yy]$$

```
f_1 := subs(simplify(JI_y*subs(f_0,SPsi_y)), [xx=x,yy=y]);
sol := solve(f_1,[x,y]);
```

$$\begin{pmatrix} -y\,x^3 - y\,x^2 + 1 \\ x\,y^2\,(x + 1) \end{pmatrix}$$

$$\emptyset$$

Daher führen wir eine Aufblasung in x-Richtung durch. Der Ursprung ist weiterhin der einzige stationäre Punkt und die Jacobi-Matrix ist immer noch nilpotent.

```
Psi_x := [ xx, xx*yy ];
SPsi_x := [ x=xx, y=xx*yy ]:
J_x := jacobian(Psi_x, [xx,yy]):
JI_x := simplify(J_x^(-1)):
```

$$[xx,\ xx\ yy]$$

```
f_1 := subs(simplify(JI_x*subs(f_0,SPsi_x)), [xx=x,yy=y]);
sol := solve(f_1,[x,y]); sol := sol[1]:
subs(jacobian(f_1,[x,y]),sol)
```

$$\begin{pmatrix} x\,y \\ -y^2 + x\,y + x \end{pmatrix}$$

$\{[x = 0, y = 0]\}$

$$\begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}$$

Wieder Bestimmung des charakteristischen Polynoms...

```
m := min(op(map(f_1,ldegree)));
fm := map(map(map(f_1,monomials), select, p -> degree(p)=m), _plus@op);
F := factor(x*fm[2]-y*fm[1])
```

1

$$\begin{pmatrix} 0 \\ x \end{pmatrix}$$

x^2

Diesmal ist x=0 die einzige charakteristische Richtung; deshalb führen wir eine y-Aufblasung durch. Nach wie vor gibt es nur den Ursprung als stationären Punkt; jetzt sogar mit Nullmatrix als Jacobi-Matrix.

```
f_2 := subs(simplify(JI_y*subs(f_1,SPsi_y)), [xx=x,yy=y]);
sol := solve(f_2,[x,y]); sol := sol[1]:
subs(jacobian(f_2,[x,y]),sol)
```

$$\begin{pmatrix} -x\,(x - 2\,y + x\,y) \\ y\,(x - y + x\,y) \end{pmatrix}$$

$\{[x = 0, y = 0]\}$

$$\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

Das charakteristische Polynom zerfällt in drei verschiedene Linearfaktoren, so daß es drei charakteristische Richtung gibt.

```
m := min(op(map(f_2,ldegree)));
fm := map(map(map(f_2,monomials), select, p -> degree(p)=m), _plus@op);
F := factor(x*fm[2]-y*fm[1])
```

2

$$\begin{pmatrix} 2\,x\,y - x^2 \\ x\,y - y^2 \end{pmatrix}$$

$x\,y\,(2\,x - 3\,y)$

Wir führen erneut eine y-Aufblasung durch. Da wir m=2 haben, kann diesmal durch y geteilt werden. Wir finden zwei stationäre Punkte, die beide Sattelpunkte sind.

```
f_3 := subs(simplify(JI_y*subs(f_2,SPsi_y)), [xx=x,yy=y]); f_3 := f_3/y;
sol := solve(f_3,[x,y]);
for s in sol do
    J := subs(jacobian(f_3,[x,y]),s); print(J);
    print(linalg::eigenvectors(J));
end_for
```

$$\begin{pmatrix} -x\,y\,(2\,x+2\,x\,y-3) \\ y^2\,(x+x\,y-1) \end{pmatrix}$$

$$\begin{pmatrix} -x\,(2\,x+2\,x\,y-3) \\ y\,(x+x\,y-1) \end{pmatrix}$$

$$\left\{[x=0,\ y=0],\ \left[x=\frac{3}{2},\ y=0\right]\right\}$$

$$\begin{pmatrix} -3 & -\frac{9}{2} \\ 0 & \frac{1}{2} \end{pmatrix}$$

$$\left[\left[-3,\ 1,\ \left[\begin{pmatrix} 1 \\ 0 \end{pmatrix}\right]\right],\ \left[\frac{1}{2},\ 1,\ \left[\begin{pmatrix} -\frac{9}{7} \\ 1 \end{pmatrix}\right]\right]\right]$$

$$\begin{pmatrix} 3 & 0 \\ 0 & -1 \end{pmatrix}$$

$$\left[\left[-1,\ 1,\ \left[\begin{pmatrix} 0 \\ 1 \end{pmatrix}\right]\right],\ \left[3,\ 1,\ \left[\begin{pmatrix} 1 \\ 0 \end{pmatrix}\right]\right]\right]$$

Eine x-Aufblasung hätte ein äquivalentes Ergebnis geliefert.

```
fx_3 := subs(simplify(JI_x*subs(f_2,SPsi_x)), [xx=x,yy=y]); fx_3 := fx_3
sol := solve(fx_3,[x,y]);
for s in sol do
    J := subs(jacobian(fx_3,[x,y]),s); print(J);
    print(linalg::eigenvectors(J));
end_for
```

$$\begin{pmatrix} -x^2\,(x\,y-2\,y+1) \\ x\,y\,(2\,x\,y-3\,y+2) \end{pmatrix}$$

$$\begin{pmatrix} -x\,(x\,y-2\,y+1) \\ y\,(2\,x\,y-3\,y+2) \end{pmatrix}$$

$$\left\{[x=0,\ y=0],\ \left[x=0,\ y=\frac{2}{3}\right]\right\}$$

$$\begin{pmatrix} \frac{1}{3} & 0 \\ \frac{8}{9} & -2 \end{pmatrix}$$

$$\left[\left[-2,\ 1,\ \left[\begin{pmatrix} 0 \\ 1 \end{pmatrix}\right]\right],\ \left[\frac{1}{3},\ 1,\ \left[\begin{pmatrix} \frac{21}{8} \\ 1 \end{pmatrix}\right]\right]\right]$$

$$\begin{pmatrix} -1 & 0 \\ 0 & 2 \end{pmatrix}$$

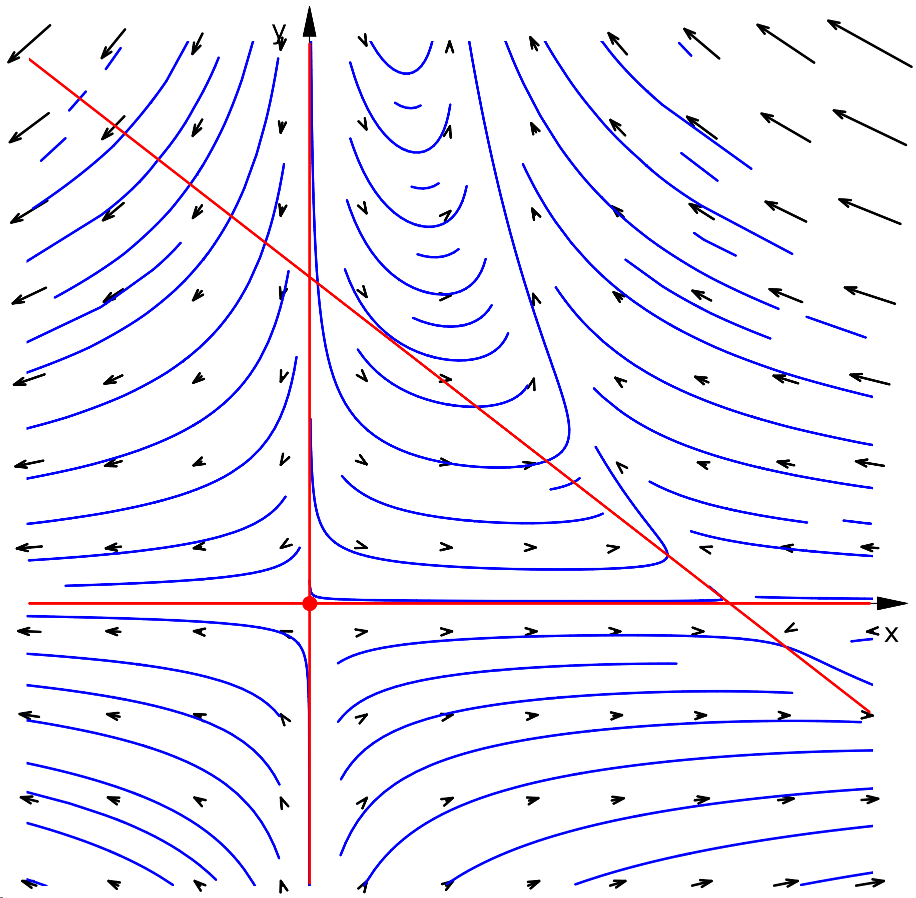
$$\left[\left[-1,\ 1,\ \left[\begin{pmatrix} 1 \\ 0 \end{pmatrix}\right]\right],\ \left[2,\ 1,\ \left[\begin{pmatrix} 0 \\ 1 \end{pmatrix}\right]\right]\right]$$

Wir zeichnen nun "rückwärts" die Phasenporträts der verschiedenen aufgeblasenen Systeme. Das letzte System hat zwei Sattelpunkte und daher wird eine hinreichend kleine Umgebung des exzeptionellen Divisors, also der x-Achse in sechs Gebiete unterteilt. Beachte, daß die roten Kurven nur lineare Approximationen der

Separatrizen darstellen.

```
a := -1: b := 2:
vf := plot::VectorField2d(f_3, x=a..b, y=a..b, Color=RGB::Black):
pp := plot::Streamlines2d(f_3, x=a..b, y=a..b, MinimumDistance=0.1,
                          Color=RGB::Blue):
eq := plot::Point2d(0,0, Color=RGB::Red, PointSize=2):
sep := plot::Line2d([3/2-9*a/7,a],[3/2-9*b/7,b], Color=RGB::Red):
sepx := plot::Line2d([a,0],[b,0], Color=RGB::Red):
sepy := plot::Line2d([0,a],[0,b], Color=RGB::Red):
```

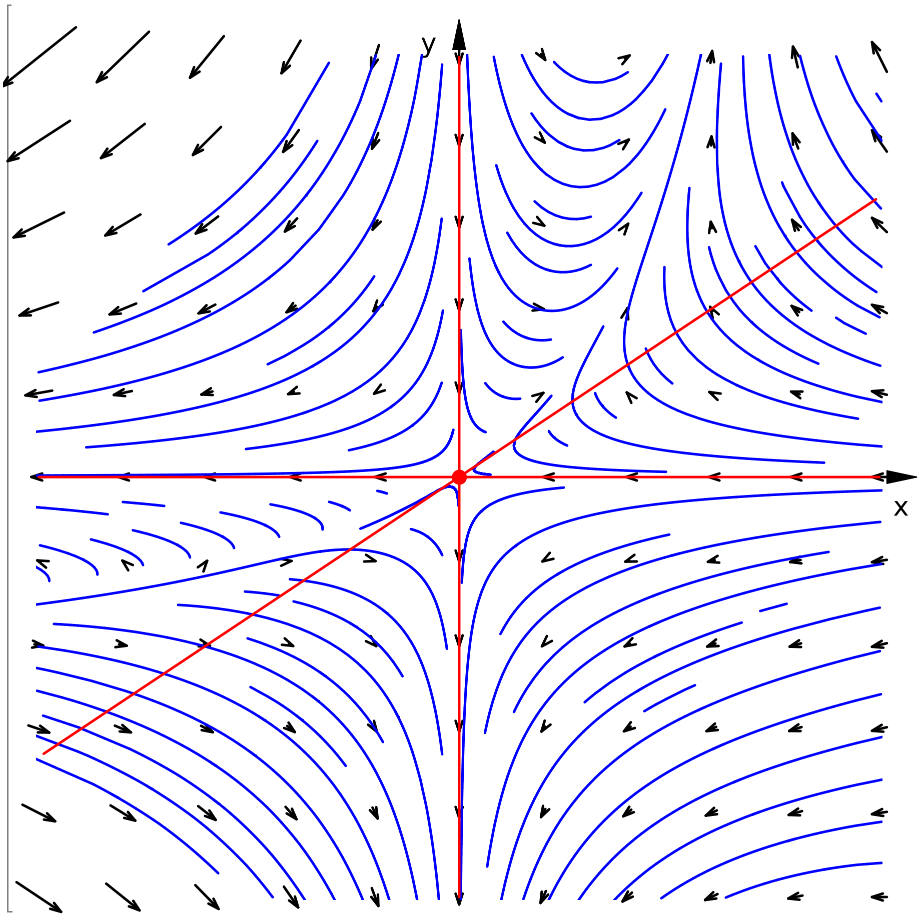
```
plot(vf,pp,eq,sep,sepx,sepy,
     ViewingBox=[a..b,a..b], Scaling=Constrained, TicksNumber=None,
     Width=120,Height=120)
```



Aus den beiden Sattelpunkten werden bei der Rücktransformation zwei (lineare Approximationen von) Separatrizen, deren Steigungen durch die x-Koordinaten der Sattelpunkte bestimmt sind. Als weitere Separatrix finden wir den exzeptionellen Divisor. Wir haben also jetzt einen stationären Punkt mit sechs Sektoren.

```
a := -2: b := 2:
vf := plot::VectorField2d(f_2, x=a..b, y=a..b, Color=RGB::Black):
pp := plot::Streamlines2d(f_2, x=a..b, y=a..b, MinimumDistance=0.1,
                          Color=RGB::Blue):
eq := plot::Point2d(0,0, Color=RGB::Red, PointSize=2):
sep := plot::Implicit2d(y-2*x/3, x=a..b, y=a..b, Color=RGB::Red):
sepx := plot::Line2d([a,0],[b,0], Color=RGB::Red):
sepy := plot::Line2d([0,a],[0,b], Color=RGB::Red):
```

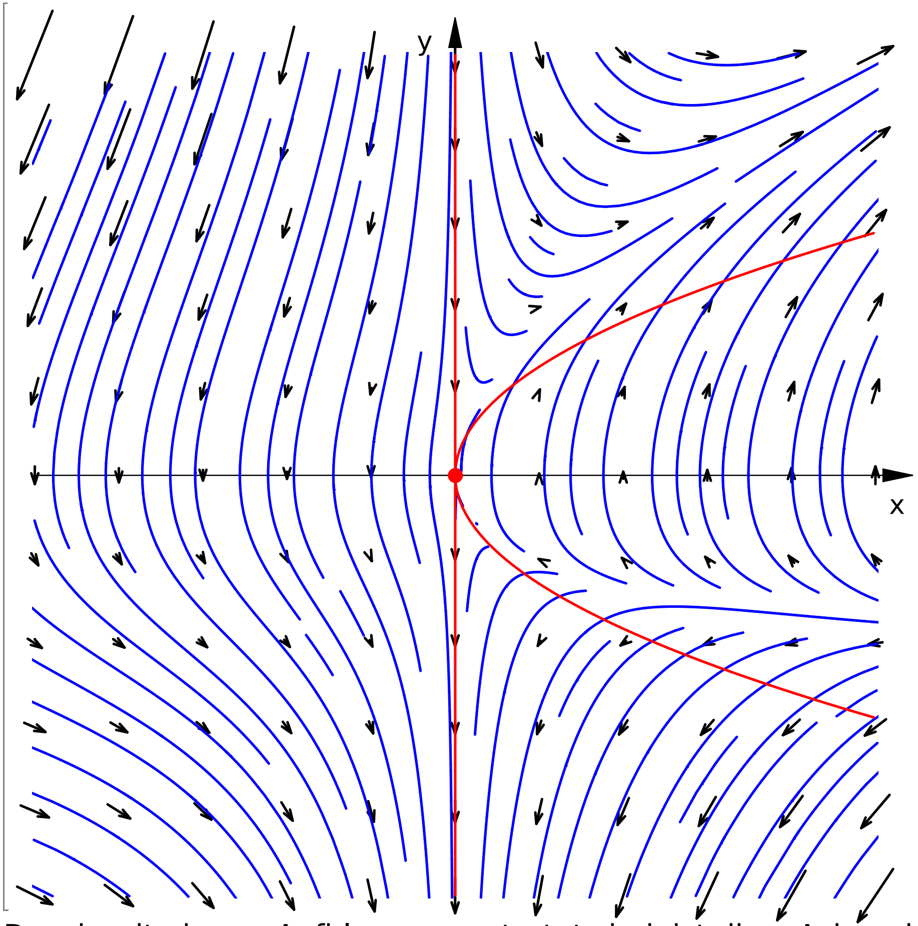
```
plot(vf,pp,eq,sep,sepx,sepy,
     ViewingBox=[a..b,a..b], Scaling=Constrained, TicksNumber=None,
     Width=120,Height=120)
```



f_2 entstand durch eine y-Aufblasung, deren exzeptioneller Divisor die x-Achse ist. Dieser verschwindet aber bei der Rücktransformation in den Ursprung. Aus der Geraden y=2/3*x wird die quadratische Kurve x=3/2*y^2 (weiterhin nur eine Approximation!). Die y-Achse wird auf sich selbst abgebildet. Wir haben daher nur noch vier Sektoren.

```
a := -2: b := 2:
vf := plot::VectorField2d(f_1, x=a..b, y=a..b, Color=RGB::Black):
pp := plot::Streamlines2d(f_1, x=a..b, y=a..b, MinimumDistance=0.1,
                          Color=RGB::Blue):
eq := plot::Point2d(0,0, Color=RGB::Red, PointSize=2):
sep := plot::Implicit2d(y^2-2*x/3, x=a..b, y=a..b, Color=RGB::Red):
sepy := plot::Line2d([0,a],[0,b], Color=RGB::Red):
```

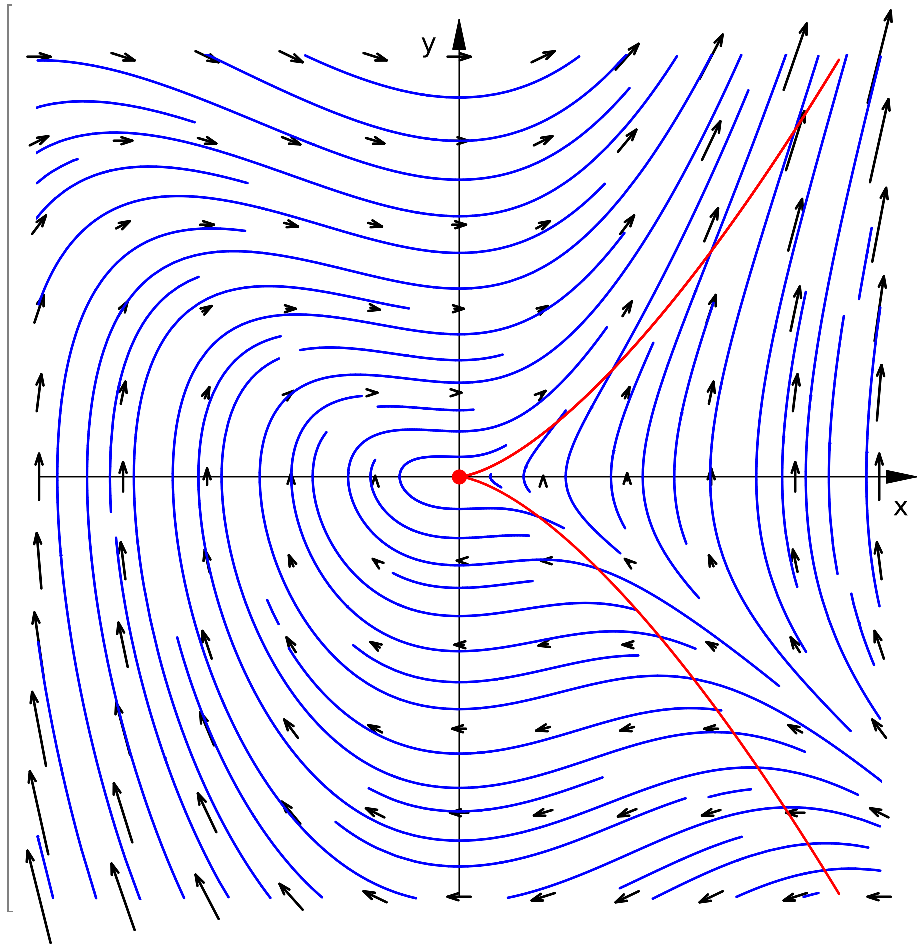
```
plot(vf,pp,eq,sep,sepy,
     ViewingBox=[a..b,a..b], Scaling=Constrained, TicksNumber=None,
     Width=120,Height=120)
```



Da wir mit einer x-Aufblasung gestartet sind, ist die y-Achse der exzeptionelle Divisor und verschwindet bei der Rücktransformation in den Ursprung. Aus der Kurve x=3/2*y^2 wird die Kurve x^3=3/2*y^2 als (Approximation der) Separatrix und haben nur noch zwei Sektoren.

```
a := -2: b := 2:
vf := plot::VectorField2d(f_0, x=a..b, y=a..b, Color=RGB::Black):
pp := plot::Streamlines2d(f_0, x=a..b, y=a..b, MinimumDistance=0.1,
                          Color=RGB::Blue):
eq := plot::Point2d(0,0, Color=RGB::Red, PointSize=2):
sep := plot::Implicit2d(y^2-2*x^3/3, x=a..b, y=a..b, Color=RGB::Red):
```

```
plot(vf,pp,eq,sep,
     ViewingBox=[a..b,a..b], Scaling=Constrained, TicksNumber=None,
     Width=120,Height=120)
```


y
x