phaseR Exercise Solutions

These solutions contain by no means the only suitable code for performing phase plane analysis upon the example systems. Moreover, again it is obviously not a useful exercise for me to provide example hand drawn plots of the nullclines and trajectories. A user should simply verify that any plots they have drawn by hand are similar to those generated by phaseR. It is also useful to note that whilst phase portrait plots are always here produced for specific parameter cases, you should be able to plot more general versions of the phase portraits yourself by hand (grouping by case where necessary). In this way, the phase portrait method can become as general as performing the Taylor approach yourself.

Exercise 1: Here, to produce the plot in Section 2.2, of the flow field and several trajectories, we use the following code:

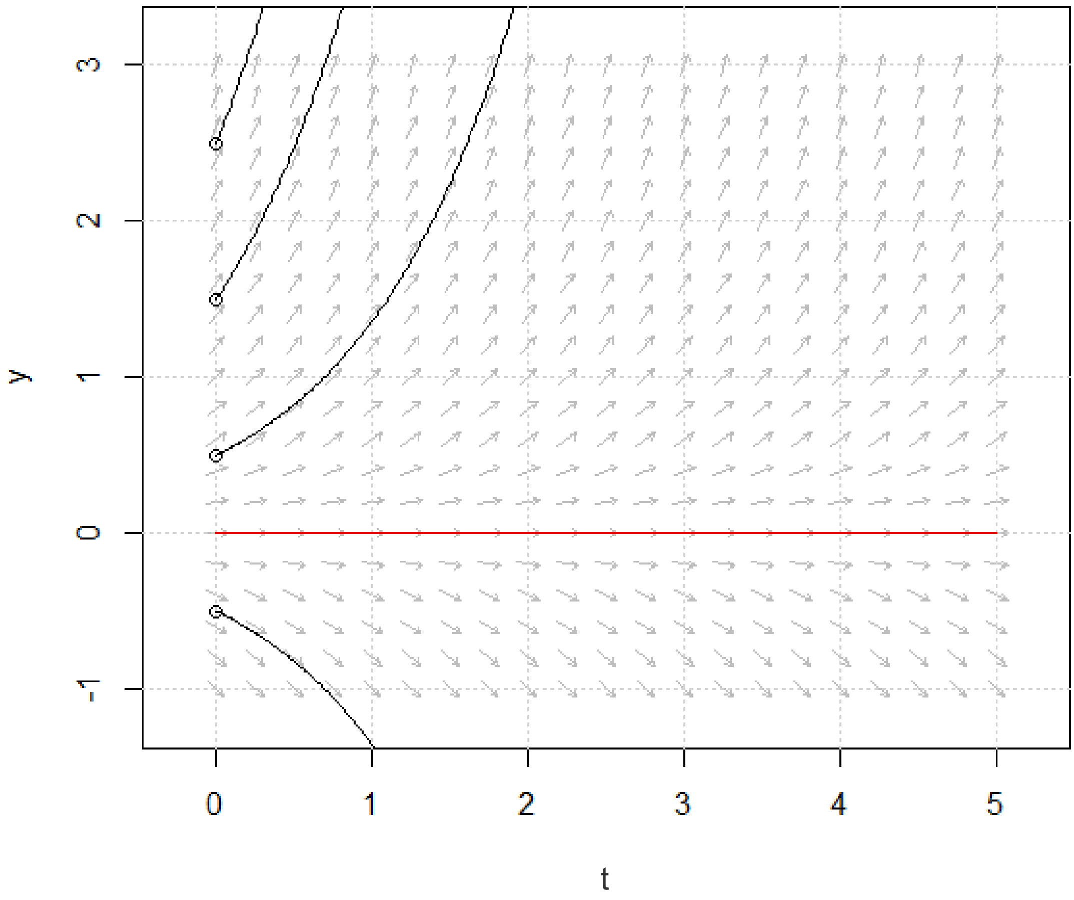
```
> example1.flowField <- flowField(example1, x.lim = c(0, 4),
+ y.lim = c(-4, 4), points = 21, system = "one.dim", add =
+ FALSE, xlab = "t", ylab = "y")
> grid()
> example1.trajectory <- trajectory(example1, y0 = c(-3, -1,
+ 0, 1, 3), t.end = 4, system = "one.dim")
```

To produce the phase portrait plot of Section 2.3, we turn to phasePortrait, employing the following code:

```
> example1.phasePortrait <- phasePortrait(example1, y.lim =
+ c(-4, 4), points = 7, xlab = "y", ylab = "f(y)")
> grid()
```

Exercise 2: To analyse the exponential model, we begin by plotting the flow field and several trajectories for the case $\beta = 1$. In addition we add a horizontal line at any equilibrium points:

```
> exponential.1.flowField <- flowField(exponential, x.lim =
+ c(0, 5), y.lim = c(-1, 3), parameters = 1, points = 21,
+ system = "one.dim", add = FALSE, xlab = "t", ylab = "y")
> grid()
> exponential.1.nullclines <- nullclines(exponential,
+ x.lim = c(0, 5), y.lim = c(-1, 3), parameters = 1,
+ system = "one.dim")
> exponential.1.trajectory <- trajectory(exponential, y0 =
+ c(-0.5, 0.5, 1.5, 2.5), t.end = 5, parameters = 1, system
= "one.dim")
```

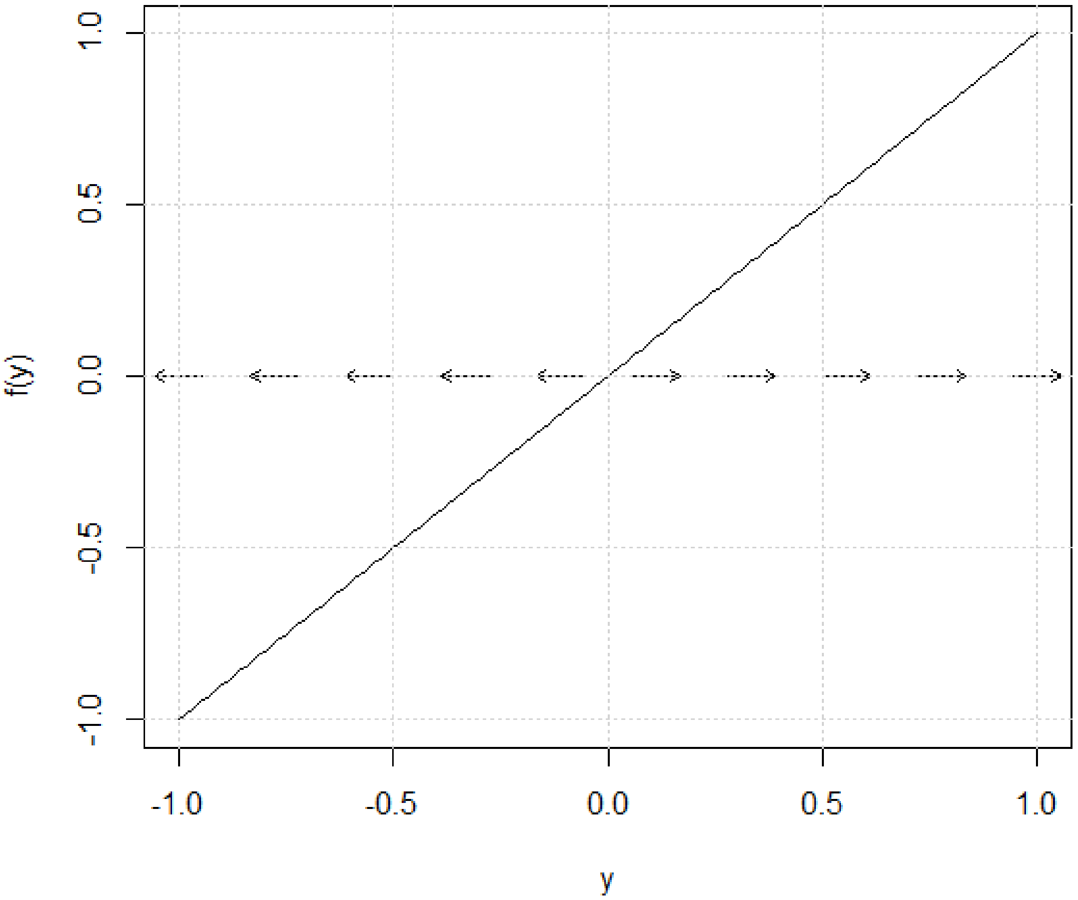


Thus one equilibrium point appears to have been found; at $y_* = 0$. We can verify this location analytically by setting the derivative to zero:

$$\beta y_* = 0 \Longrightarrow y_* = 0.$$

To assess stability for $\beta = 1$ we first plot the phase portrait:

```
> exponential.1.phasePortrait <- phasePortrait(exponential,
+ y.lim = c(-1, 1), parameters = 1, points = 10, xlab = "y",
+ ylab = "f(y)")
> grid()
```



Thus, $y_* = 0$ will be unstable here. In addition, we can use perturbation analysis to classify $y_* = 0$ in the general case:

$$\left(\frac{df}{dy}\right)\bigg|_{y=y_*} = \beta.$$

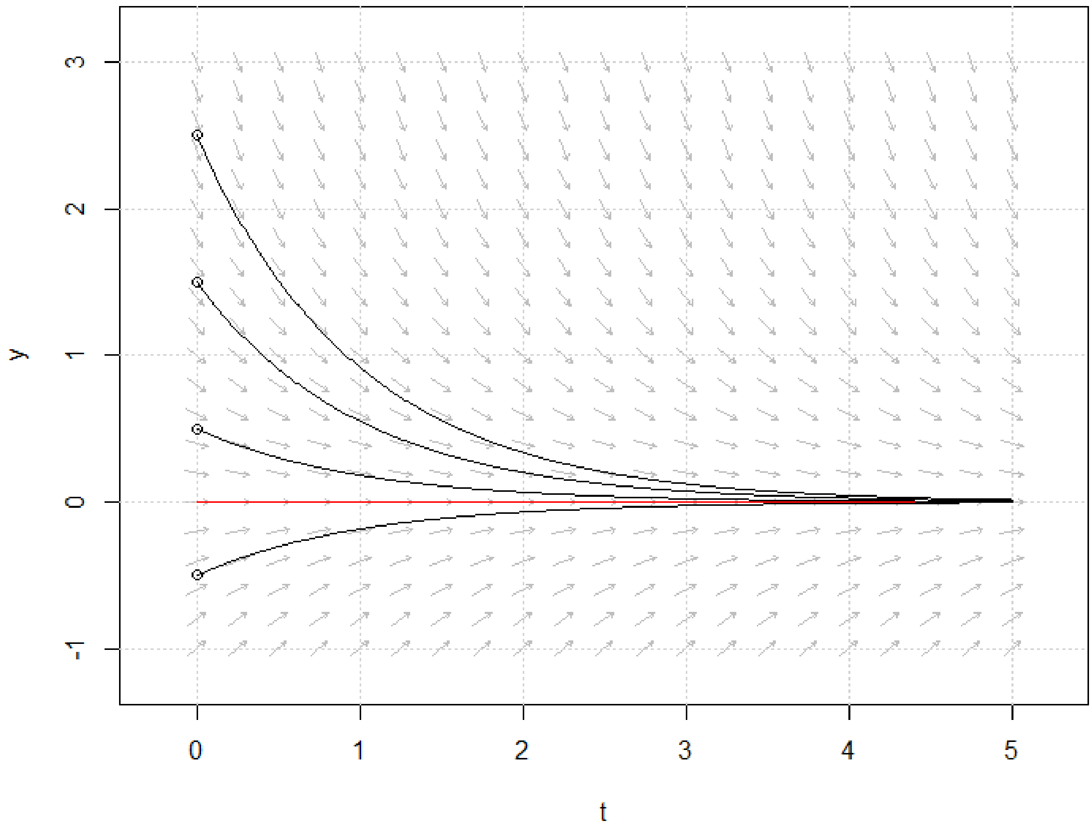
Thus the equilibrium point will be stable if $\beta < 0$, but unstable if $\beta > 0$. We can verify that $\beta = 1$ results in $y_* = 0$ being unstable using `stability`:

```
> exponential.1.stability <- stability(exponential, y.star =
+ 0, parameters = 1, system = "one.dim")
Discriminant: 1   Classification: Unstable
```

Thus both methods have drawn the same conclusion; for $\beta = 1$, $y_* = 0$ is unstable. Indeed, this is clearly evident from the flow field and trajectories we plotted above as well.

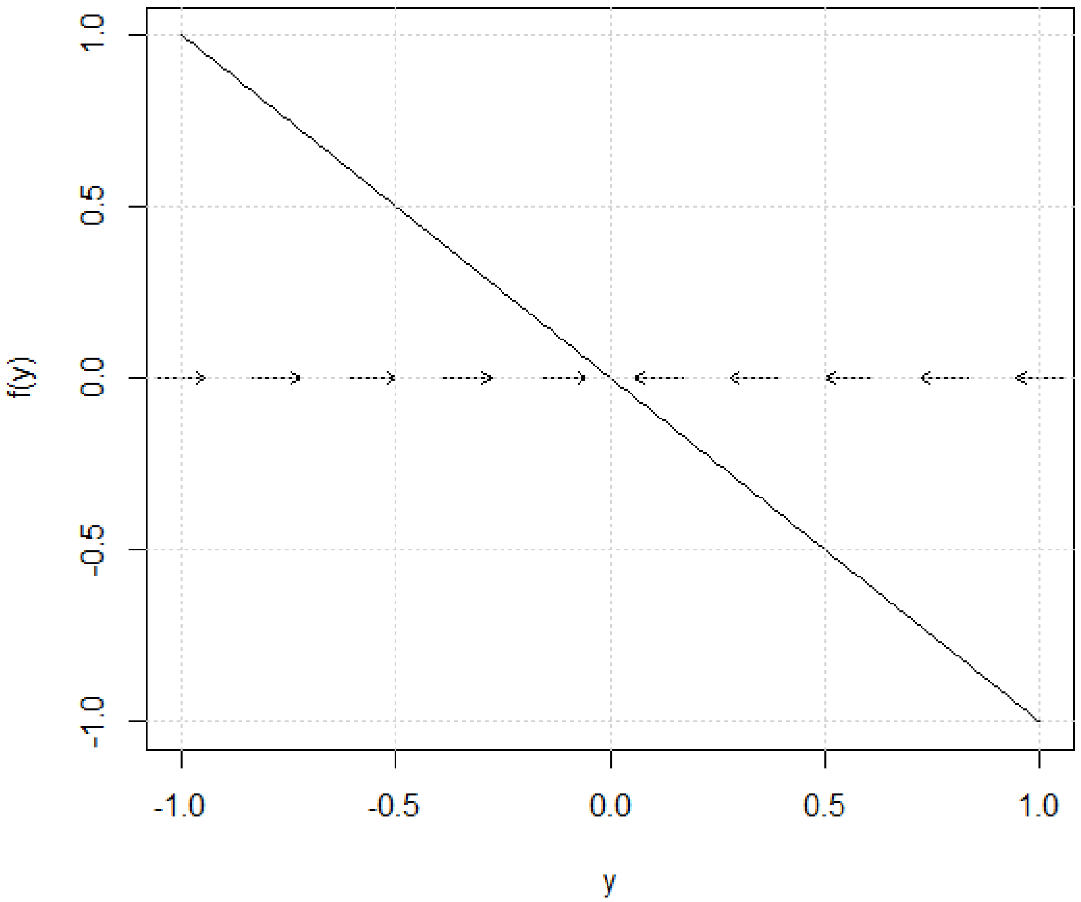
We now further focus on the case $\beta < 0$, taking $\beta = -1$ as an example to plot the flow field and trajectories:

```
> exponential.2.flowField <- flowField(exponential, x.lim =
+ c(0, 5), y.lim = c(-1, 3), parameters = -1, points = 21,
+ system = "one.dim", add = FALSE, xlab = "t", ylab = "y")
> grid()
> exponential.2.nullclines <- nullclines(exponential,
+ x.lim = c(0, 5), y.lim = c(-1, 3), parameters = -1,
+ system = "one.dim")
> exponential.2.trajectory <- trajectory(exponential, y0 =
+ c(-0.5, 0.5, 1.5, 2.5), t.end = 5, parameters = -1,
+ system = "one.dim")
```



Thus, $y_* = 0$ is again the only equilibrium point (as we calculated earlier). Moreover, we observed earlier that $y_* = 0$ will, for $\beta = -1$, be stable. Whilst this is evident from the above plot, we can check it by first plotting the phase portrait:

```
> exponential.2.phasePortrait <- phasePortrait(exponential,
+ y.lim = c(-1, 1), parameters = -1, points = 10, xlab =
+ "y", ylab = "f(y)")
> grid()
```



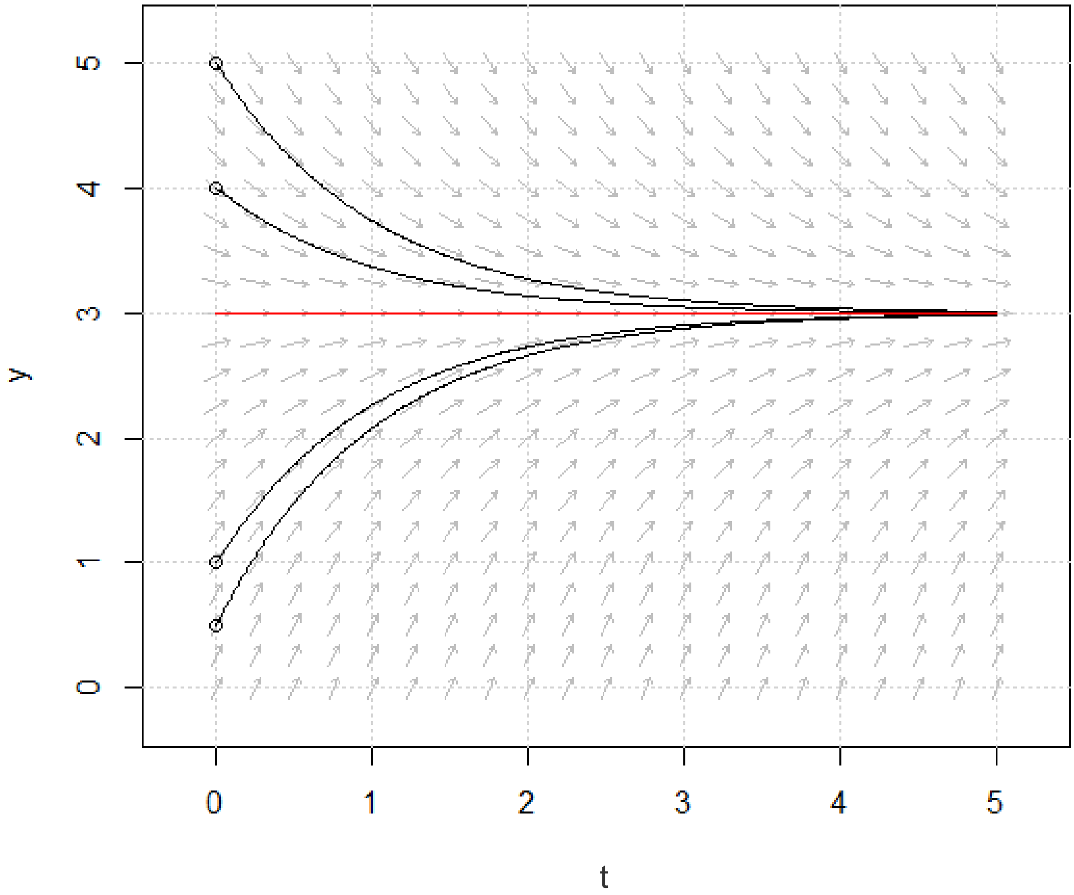
In addition, we can use `stability` and the Taylor approach:

```
> exponential.2.stability <- stability(exponential, y.star =
+ 0, parameters = -1, system = "one.dim")
Discriminant: -1   Classification: Stable
```

Thus, in general we have seen that the exponential model has only one equilibrium point, and its stability is entirely determined by the sign of β. Biologically, the sign of β here would reflect whether we expected the species under study to grow or decay. So, for example, a rapidly growing population of worms would have $\beta > 0$, whilst a model of radioactive decay would have $\beta < 0$.

Exercise 3: We begin by plotting the flow field, trajectories and identifying the equilibrium point for the suggested case $\beta = 1, K = 3$:

```
> monomolecular.1.flowField <- flowField(monomolecular,
+ x.lim = c(0, 5), y.lim = c(0, 5), parameters = c(1, 3),
+ points = 21, system = "one.dim", add = FALSE, xlab = "t",
+ ylab = "y")
> grid()
> monomolecular.1.nullclines <- nullclines(monomolecular,
+ x.lim = c(0, 5), y.lim = c(0, 5), parameters = c(1, 3),
+ points = 1000, system = "one.dim")
> monomolecular.1.trajectory <- trajectory(monomolecular,
+ y0 = c(0.5, 1, 4, 5), t.end = 5, parameters = c(1, 3),
+ system = "one.dim")
```

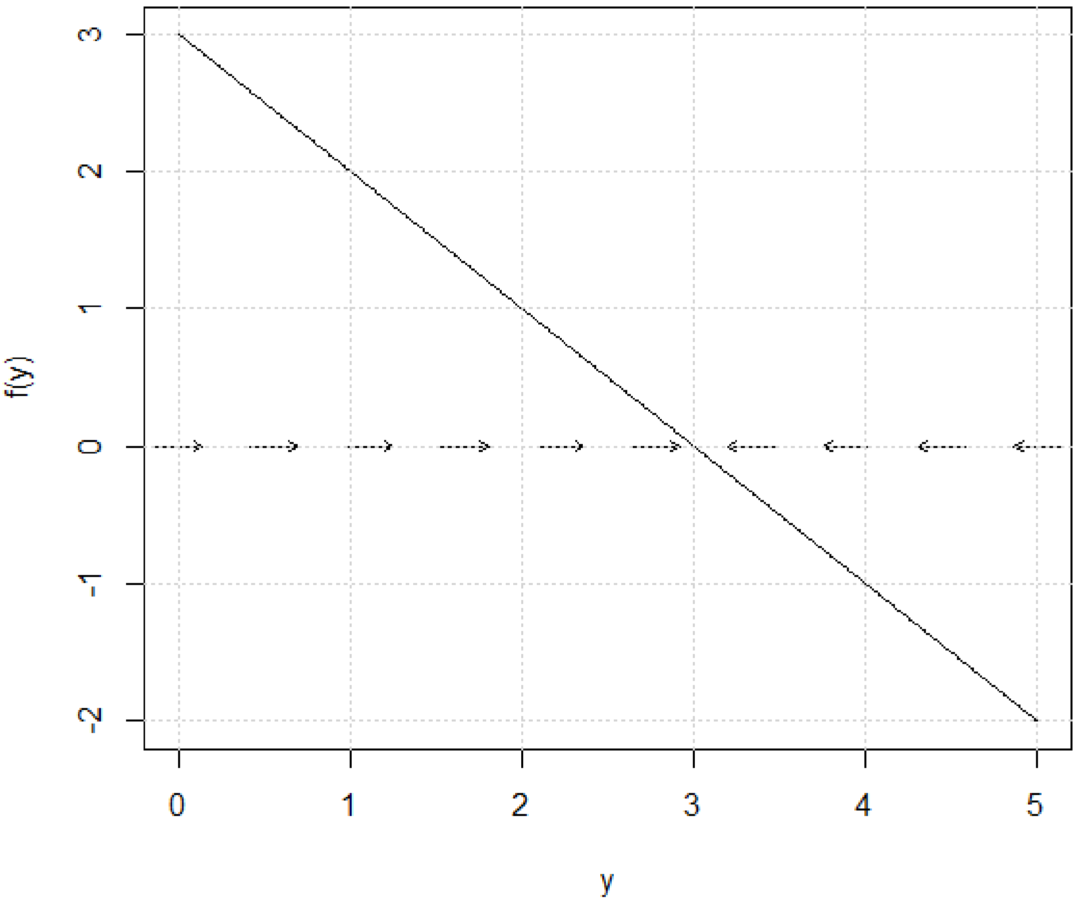


Thus, there appears to be one equilibrium point at $y_* = 3$. Setting the derivative to zero we can check this analytically:

$$\beta(K - y_*) = 0 \Longrightarrow y_* = K,$$

which would indeed translate to $y_* = 3$ in our case. To check the stability of $y_* = 3$ here we begin by plotting the phase portrait:

```
> monomolecular.1.phasePortrait <-
+ phasePortrait(monomolecular, y.lim = c(0, 5), parameters =
+ c(1, 3), points = 10, xlab = "y", ylab = "f(y)")
> grid()
```



Alternatively, we can employ perturbation analysis:

$$\left.\left(\frac{df}{dy}\right)\right|_{y=y_*} = -\beta.$$

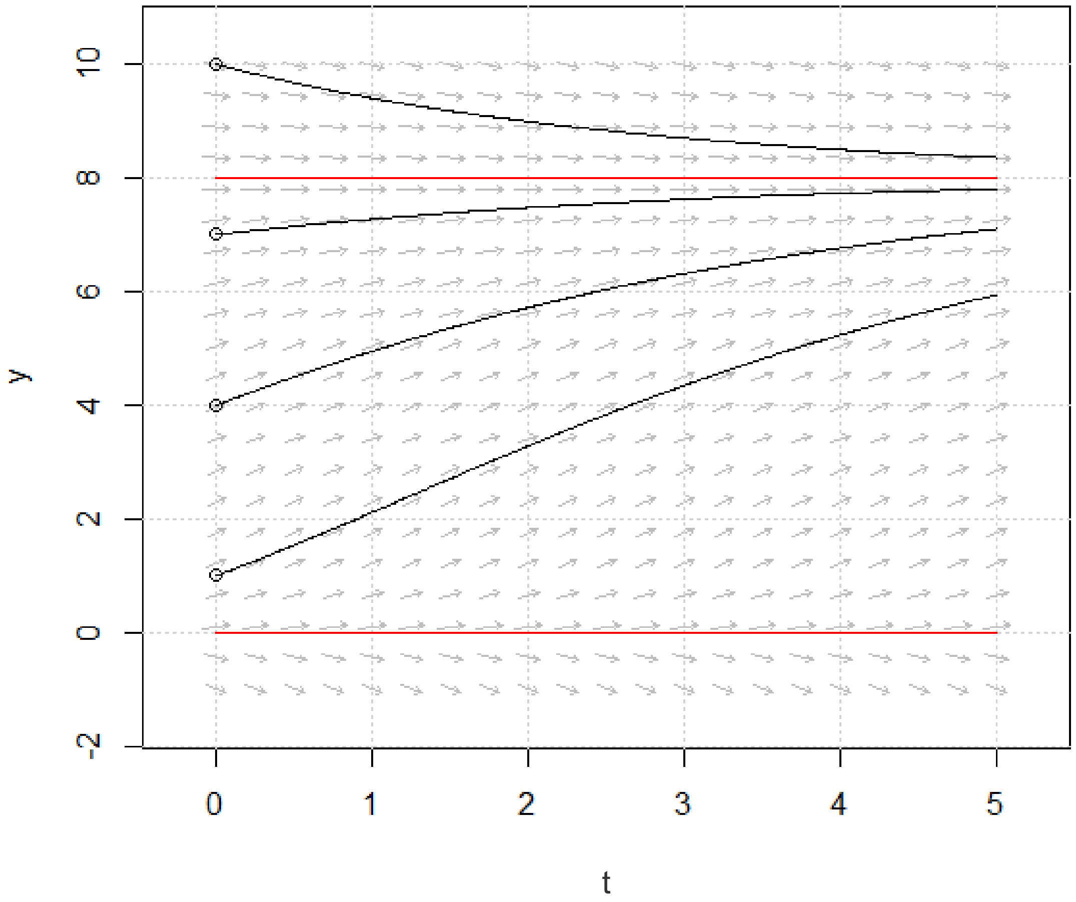
Thus for this model the equilibrium point will be stable if $\beta > 0$, but unstable if $\beta < 0$. So for our case $\beta = 1$, both methods confirm that the equilibrium point $y_* = 3$ is stable. We can also verify this using `stability`:

```
> monomolecular.1.stability <- stability(monomolecular,
+ y.star = 3, parameters = c(1, 3), system = "one.dim")
Discriminant: -1   Classification: Stable
```

Thus, as for the exponential, the monomolecular model has one equilibrium point, whose stability is entirely determined by the sign of β.

Exercise 4: To begin with, we plot the flow field, several trajectories, and identify the equilibria for the case $\alpha = 2, \beta = 1$:

```
> vonBertalanffy.flowField <- flowField(vonBertalanffy,
+ x.lim = c(0, 5), y.lim = c(-1, 10), parameters = c(2, 1),
+ points = 21, system = "one.dim", add = FALSE, xlab = "t",
+ ylab = "y")
> grid()
> vonBertalanffy.nullclines <- nullclines(vonBertalanffy, +
x.lim = c(0, 5), y.lim = c(-1, 10), parameters = c(2, 1), +
points = 1000, system = "one.dim")
> vonBertalanffy.trajectory <- trajectory(vonBertalanffy, +
y0 = c(1, 4, 7, 10), t.end = 5, parameters = c(2, 1),
+ system = "one.dim")
```



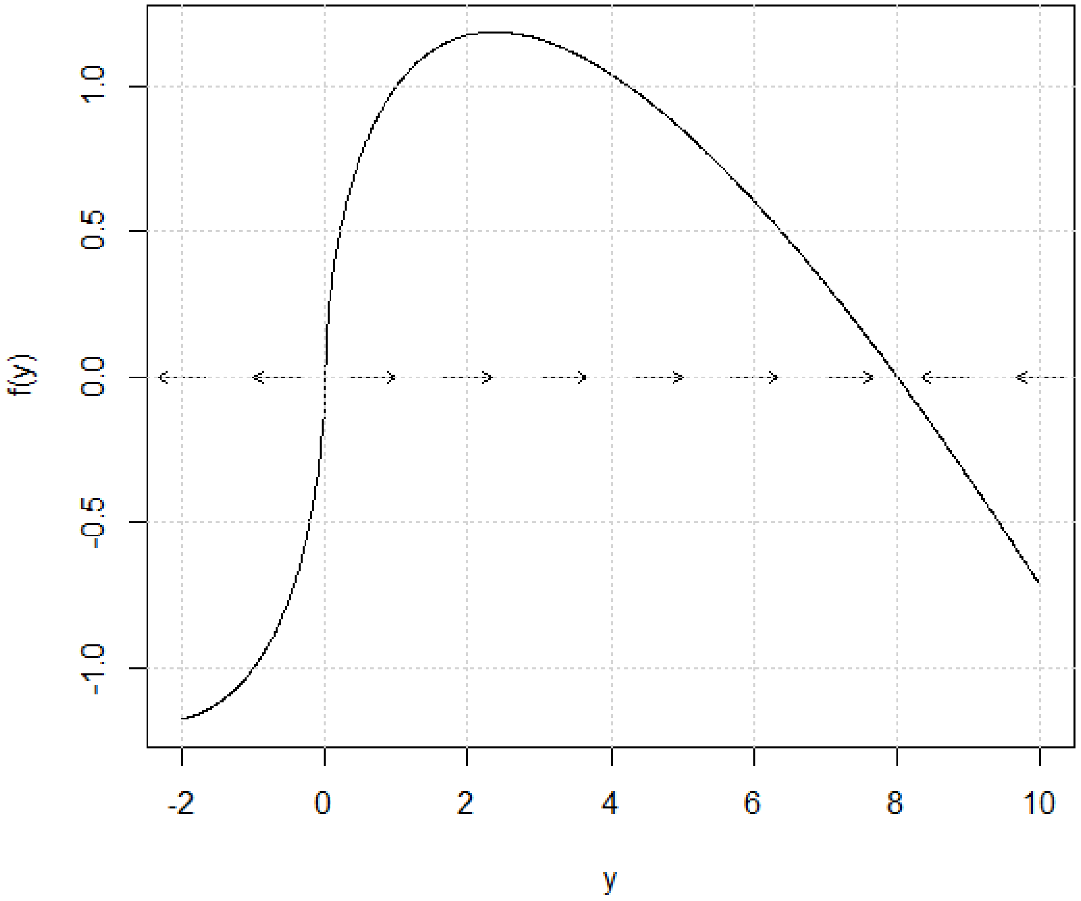
It appears there are two equilibria for this model; $y_* = 0$ and $y_* = 8$. We can find the equilibrium point's general location by setting the derivative to zero:

$$\alpha y_*^{2/3} - \beta y_* = 0 \Rightarrow y_*^{2/3}\left(\alpha - \beta y_*^{1/3}\right) = 0 \Rightarrow y_* = 0,\ \left(\frac{\alpha}{\beta}\right)^3.$$

Clearly, in our case this would correspond to $y_* = 0$ and $y_* = 8$.

As always, we can check stability using the phase portrait for our specific case:

```
> vonBertalanffy.phasePortrait <-
+ phasePortrait(vonBertalanffy, y.lim = c(-2, 10),
+ parameters = c(2, 1), points = 10, xlab = "y", ylab =
+ "f(y)")
> grid()
```



So it seems $y_* = 0$ will be unstable, but $y_* = 8$ stable.

Alternatively, we can use perturbation analysis:

$$\left(\frac{df}{dy}\right)\Bigg|_{y=y_*} = \frac{2}{3}\alpha y_*^{-1/3} - \beta.$$

Thus, if $y_* = (\alpha/\beta)^3$ then we have:

$$\left(\frac{df}{dy}\right)\Bigg|_{y=y_*} = \frac{2}{3}\alpha\left[\left(\frac{\alpha}{\beta}\right)^3\right]^{-1/3} - \beta = \frac{2}{3}\alpha\left(\frac{\beta}{\alpha}\right) - \beta = -\frac{1}{3}\beta.$$

So provided $\beta > 0$ (which should always be the case) this point will be stable. We face a problem however for the case $y_* = 0$ since $y_*^{-1/3}$ will be undefined. However, we can make a somewhat hand waving argument that provided $\alpha > 0$ (which, again, it should always be):

$$\left(\frac{df}{dy}\right)\bigg|_{y=y_*} \longrightarrow \infty \text{ as } y_* \longrightarrow 0.$$

Thus making $y_* = 0$ unstable. In spite of this, the phase portrait may be a better method here.

We can check our results using `stability` and verify our conclusions above:

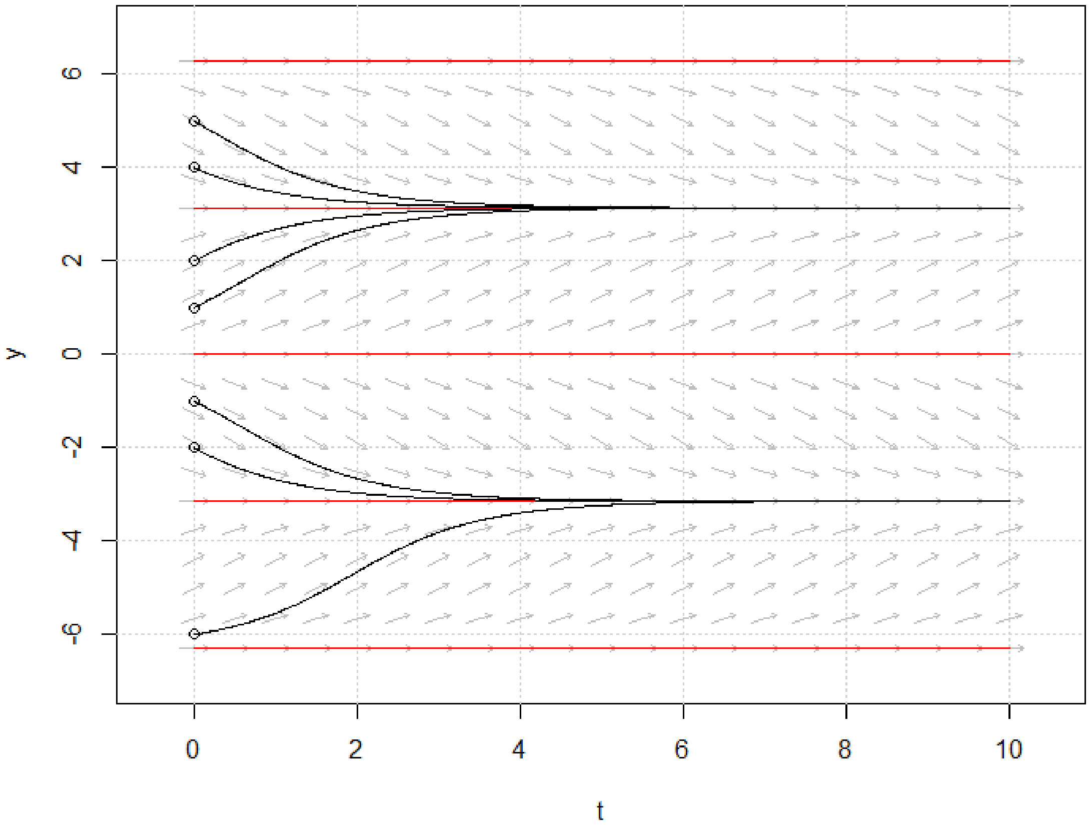
```
> vonBertalanffy.stability.1 <- stability(vonBertalanffy,
+ y.star = 0, parameters = c(2, 1), system = "one.dim")
Discriminant: 429.8869   Classification: Unstable
> vonBertalanffy.stability.2 <- stability(vonBertalanffy,
+ y.star = 8, parameters = c(2, 1), system = "one.dim")
Discriminant: -0.3333333   Classification: Stable
```

Exercise 5: We first produce a derivative function for this exercise:

```
exercise5 <- function(t, y, parameters){
  dy <- sin(y)
  list(dy)
}
```

From this, we can plot the velocity field, identify the equilbria and add several trajectories for the requested range:

```
> exercise5.flowField <- flowField(exercise5, x.lim = c(0,
+ 10), y.lim = c(-2*pi, 2*pi), points = 21, system =
+ "one.dim", add = FALSE, xlab = "t", ylab = "y")
> grid()
> exercise5.nullclines <- nullclines(exercise5, x.lim = c(0,
+ 10), y.lim = c(-2*pi - 1, 2*pi + 1), points = 100,
+ system = "one.dim")
> exercise5.trajectory <- trajectory(exercise5, y0 = c(-6,
+ -2, -1, 1, 2, 4, 5), t.end = 10, system = "one.dim")
```

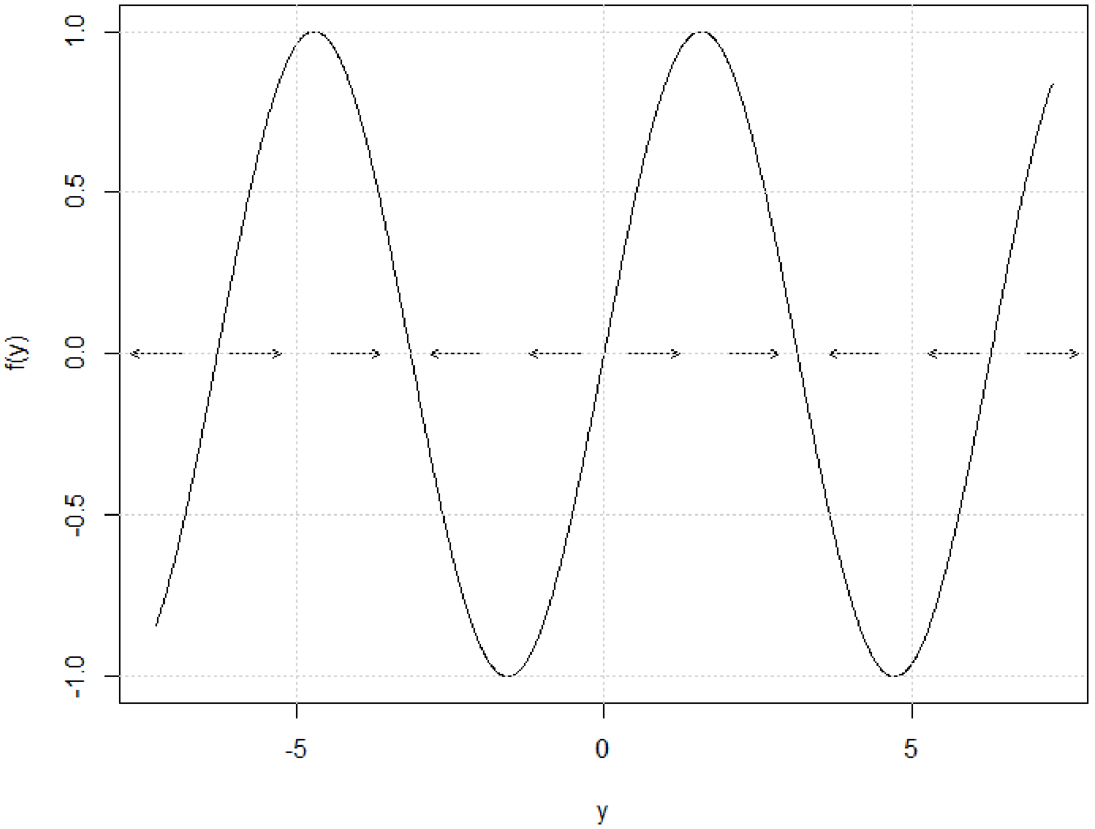


It appears we have located five equilibria; seemingly at the points $y_* = \pi n$ for n an integer. We can verify this by setting the derivative to zero:

$$\sin y_* = 0 \Longrightarrow y_* = \pi n \quad n \in \mathbb{N},$$

as expected. To determine their stability we can first use the phase portrait:

```
> exercise5.phasePortrait <- phasePortrait(exercise5,
+ y.lim = c(-2*pi - 1, 2*pi + 1), points = 10, xlab = "y",
+ ylab = "f(y)")
> grid()
```



This suggests that the points $y_* = 2\pi n$ will be unstable, but $y_* = (2n+1)\pi$ will be stable. We can verify this using the Taylor approach:

$$\left(\frac{df}{dy}\right)\bigg|_{y=y_*} = \cos y_* \begin{cases} 1: & y_* = 2\pi n, \\ -1: & y_* = (2n+1)\pi. \end{cases}$$

Finally, we use stability to demonstrate this for the points $y_* = 0$ and $y_* = \pi$ as examples:

```
> exercise5.stability.1 <- stability(exercise5, y.star = 0,
+ system = "one.dim")
Discriminant: 1   Classification: Unstable
> exercise5.stability.2 <- stability(exercise5, y.star = pi,
+ system = "one.dim")
Discriminant: -1   Classification: Stable
```

Exercise 6: To reproduce the velocity field, nullcline and trajectory plot, we can use the following code:

```
> example.2d.flowField <- flowField(lotkaVolterra, x.lim =
+ c(-1, 5), y.lim = c(-1, 5), parameters = c(1, 1, 1, 1),
+ points = 19, add = FALSE, xlab = "x", ylab = "y")
> grid()
> example.2d.nullclines <- nullclines(lotkaVolterra, x.lim =
+ c(-1, 5), y.lim = c(-1, 5), parameters = c(1, 1, 1, 1))
> y0 <- matrix(c(1, 2, 2, 2, 3, 4, -1, 4, -0.5, -1, 0.25,
+ -1), ncol = 2, nrow = 6, byrow = TRUE)
> example.2d.trajectory <- trajectory(lotkaVolterra, y0 =
+ y0, t.end = 10, parameters = c(1, 1, 1, 1), xlab = "x",
+ ylab = "y")
```

To perform the classification of the equilibria, we employ `stability`:

```
> example.2d.stability.1 <- stability(lotkaVolterra,
+ y.star = c(0, 0), parameters = c(1, 1, 1, 1))
T: 0   Delta: -1   Discriminant: 4   Classification: Saddle
> example.2d.stability.2 <- stability(lotkaVolterra,
+ y.star = c(1, 1), parameters = c(1, 1, 1, 1))
T: 0   Delta: 1   Discriminant: -4   Classification: Centre
focus
```

Finally, we can plot the dependent variables against the independent using `numericalSolution`:

```
> example.2d.numericalSolution <-
+ numericalSolution(lotkaVolterra, y0 = c(3, 4), t.end = 25,
+ parameters = c(1, 1, 1, 1), type = "one", colour =
+ c("red", "blue"))
> legend("topright", legend = c("x", "y"), lty = c(1, 1),
+ col = c("red", "blue"))
```

Exercise 7: We now proceed through each of these example linear two dimensional autonomous ODE systems:

a) Firstly, we identify the nullclines by setting the derivatives to zero:

$$\begin{aligned} x &: \; -x = 0 \Longrightarrow x = 0, \\ y &: \; -4x = 0 \Longrightarrow x = 0. \end{aligned}$$

Thus the nullclines are here the same. This means we have a line of equilibrium points given by $x = 0$. Examining the Jacobian, we see this is because we have a singular case where $\det \mathbf{J} = 0$:

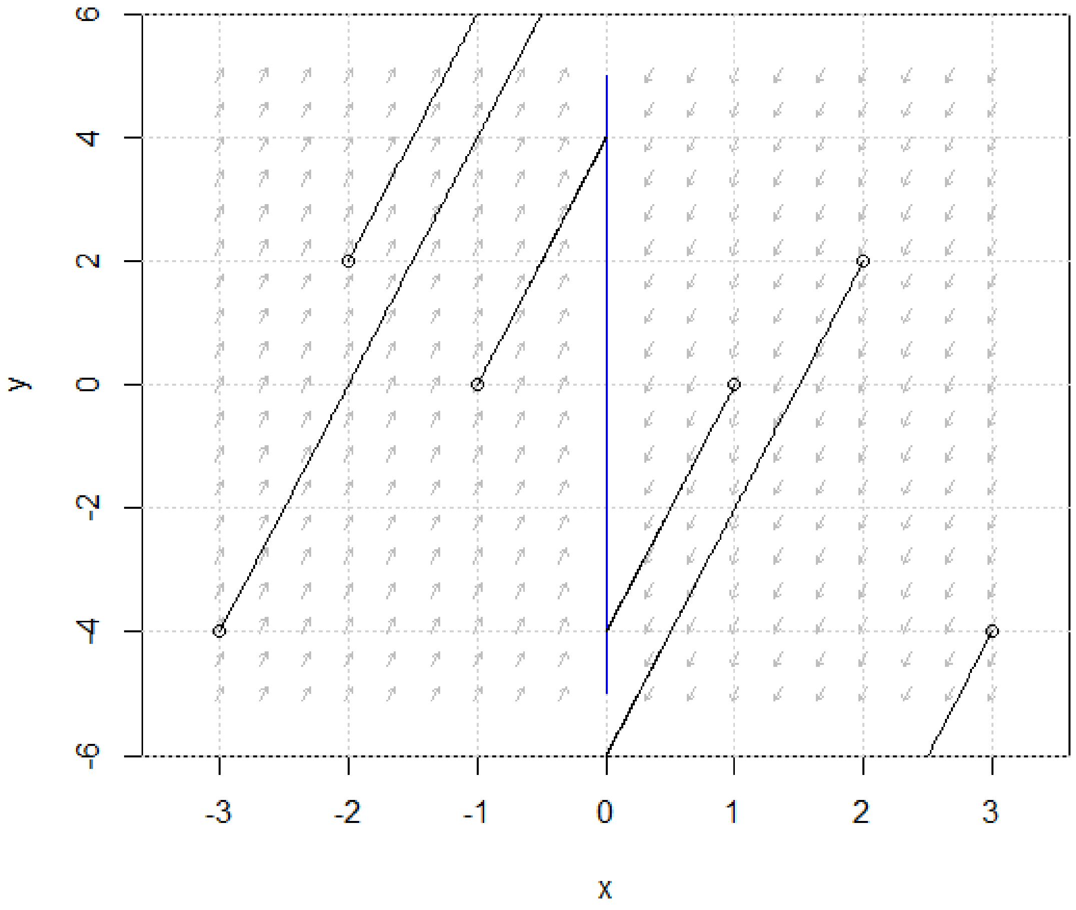
$$\mathbf{J} = \begin{pmatrix} -1 & 0 \\ -4 & 0 \end{pmatrix}.$$

Here, if we use `stability` we will be unable to classify the equilibrium points along $x = 0$ (taking (0,0) as an example):

```
> example3.stability <- stability(example3, y.star = c(0,
+ 0))
T: -1   Delta: 0   Discriminant: 1   Classification:
Indeterminate
```

However, if we plot the velocity field and several trajectories, we can identify that the equilibrium points are stable:

```
> example3.flowField <- flowField(example3, x.lim = c(-3,
+ 3), y.lim = c(-5, 5), points = 19, add = FALSE, xlab =
+ "x", ylab = "y")
> grid()
> example3.nullclines <- nullclines(example3, x.lim = c(-3,
+ 3), y.lim = c(-5, 5))
> y0 <- matrix(c(1, 0, -1, 0, 2, 2, -2, 2, 3, -4, -3, -4),
+ ncol = 2, nrow = 6, byrow = TRUE)
> example3.trajectory <- trajectory(example3, y0 = y0,
+ t.end = 10,  xlab = "x", ylab = "y")
```

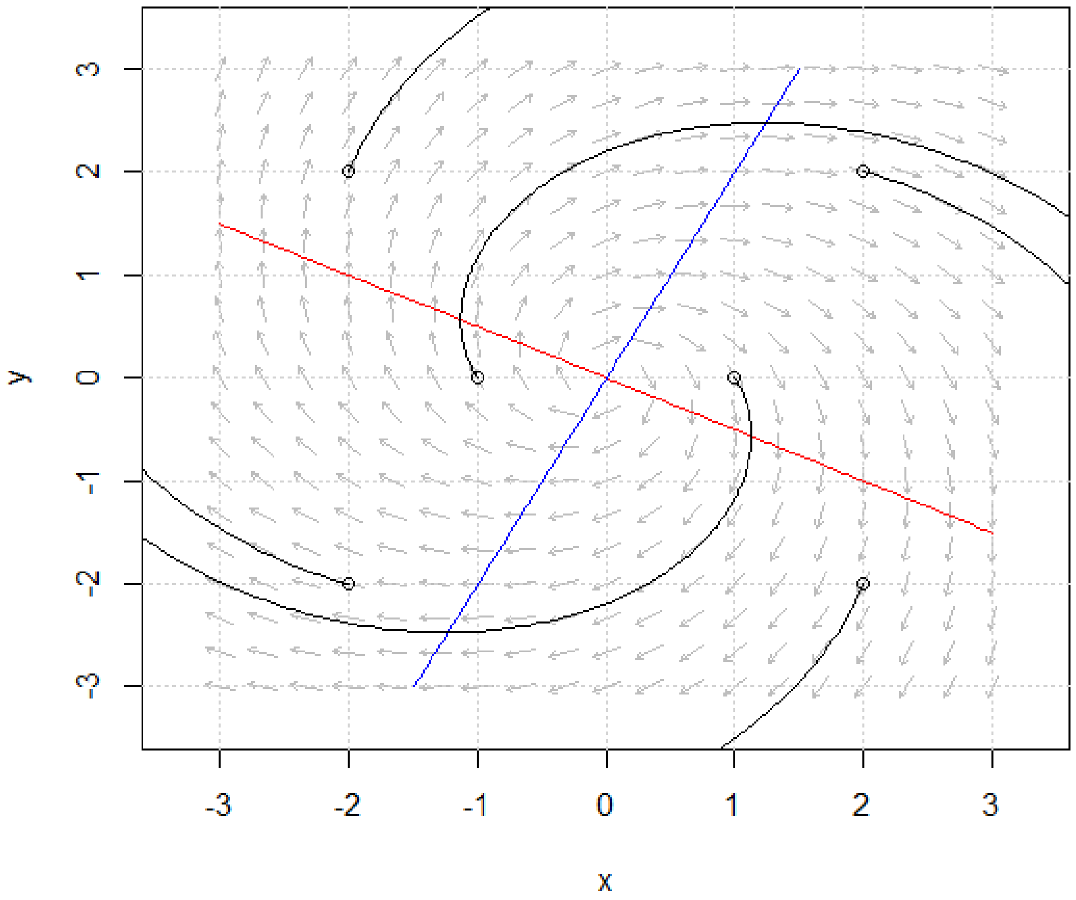


b) We begin by identifying the nullclines by setting the derivatives to zero:

$$x: \; x + 2y = 0 \Longrightarrow y = -x/2,$$
$$y: \; -2x + y = 0 \Longrightarrow y = 2x.$$

We now plot the velocity field, nullclines and several trajectories:

```
> example6.flowField <- flowField(example6, x.lim = c(-3,
+ 3), y.lim = c(-3, 3), points = 19, add = FALSE, xlab =
+ "x", ylab = "y")
> grid()
> example6.nullclines <- nullclines(example6, x.lim = c(-3,
+ 3), y.lim = c(-3, 3))
> y0 <- matrix(c(1, 0, -1, 0, 2, 2, -2, 2, 2, -2, -2, -2),
+ ncol = 2, nrow = 6, byrow = TRUE)
> example6.trajectory <- trajectory(example6, y0 = y0,
+ t.end = 10, xlab = "x", ylab = "y")
```



Now from the equations of the nullclines or from the plot we can see that the (only) equilibrium point is at (0,0), and that it appears to be an unstable focus.

To confirm this we use the Jacobian:

$$\mathbf{J} = \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix},$$
$$\Longrightarrow T = 2, \Delta = 5, T^2 - 4\Delta = -16.$$

Thus (0,0) is an unstable focus. We can also perform this analysis using `stability`:

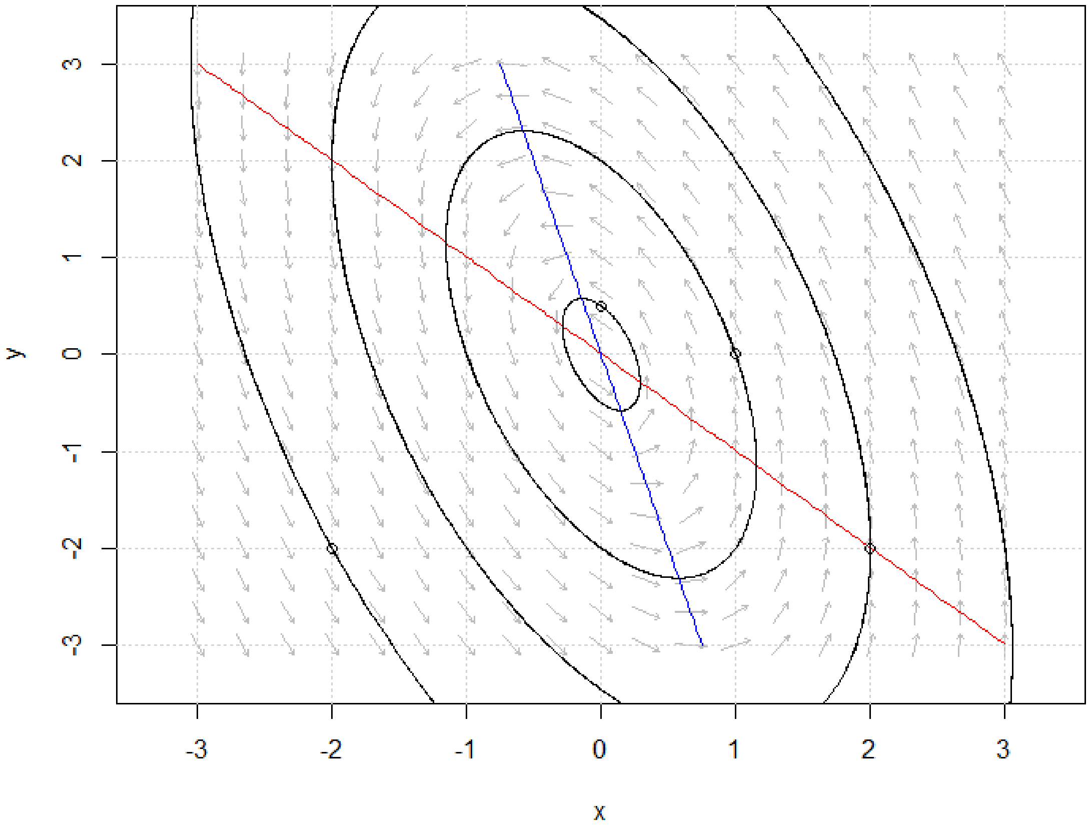
```
> example6.stability <- stability(example6, y.star = c(0,
+ 0))
T: 2   Delta: 5   Discriminant: -16   Classification:
Unstable focus
```

c) We begin by identifying the nullclines by setting the derivatives to zero:

$$x: \ -x - y = 0 \Longrightarrow y = -x,$$
$$y: \ \ 4x + y = 0 \Longrightarrow y = -4x.$$

We now plot the velocity field, nullclines and several trajectories:

```
> example7.flowField <- flowField(example7, x.lim = c(-3,
+ 3), y.lim = c(-3, 3), points = 19, add = FALSE, xlab =
+ "x", ylab = "y")
> grid()
> example7.nullclines <- nullclines(example7, x.lim = c(-3,
+ 3), y.lim = c(-3, 3))
> y0 <- matrix(c(1, 0, 0, 0.5, 2, -2, -2, -2), ncol = 2,
+ nrow = 4, byrow = TRUE)
> example7.trajectory <- trajectory(example7, y0 = y0,
+ t.end = 10, xlab = "x", ylab = "y")
```



Now from the equations of the nullclines or from the plot we can see that the (only) equilibrium point is at (0,0), and that it appears to be a centre. To confirm this we use the Jacobian:

$$\mathbf{J} = \begin{pmatrix} -1 & -1 \\ 4 & 1 \end{pmatrix},$$
$$\Rightarrow T = 0, \Delta = 3, T^2 - 4\Delta = -12.$$

Thus (0,0) is a centre. We can also perform this analysis using `stability`:

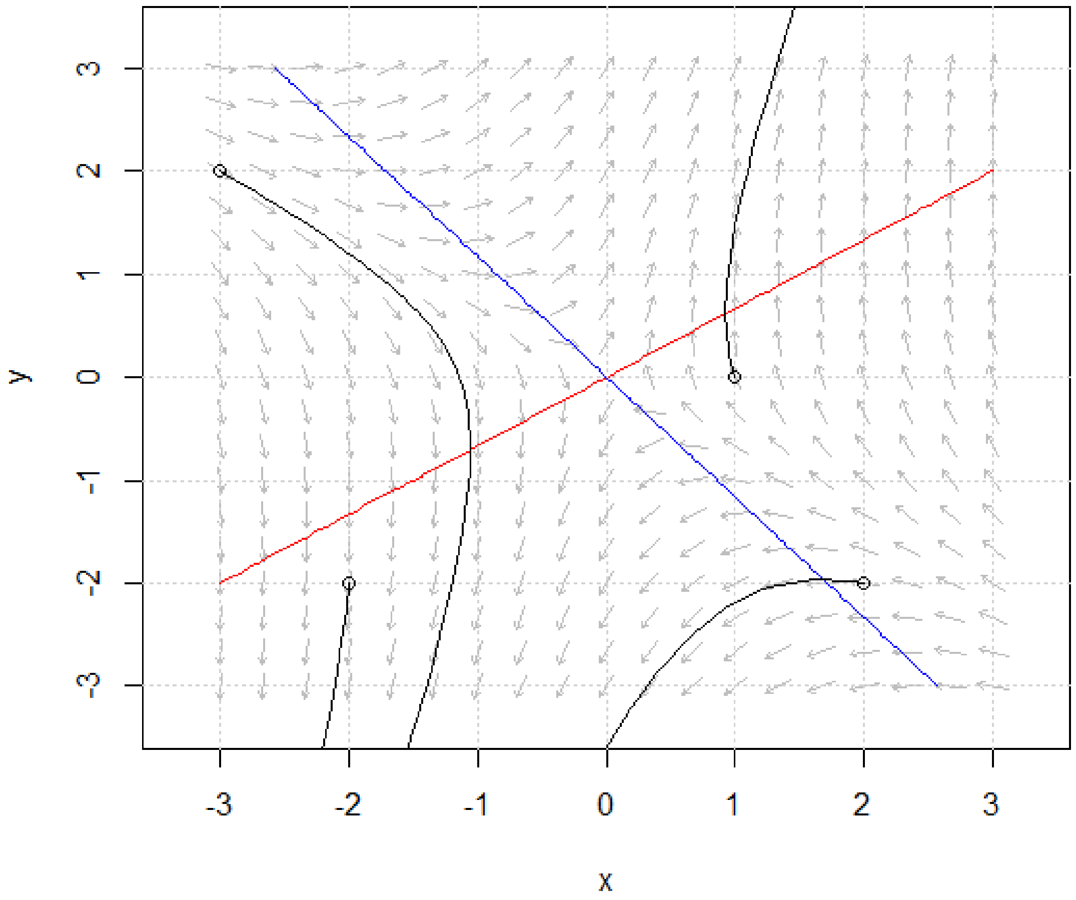
```
> example7.stability <- stability(example7, y.star = c(0,
+ 0))
T: 0   Delta: 3   Discriminant: -12   Classification: Centre
focus
```

d) We begin by identifying the nullclines by setting the derivatives to zero:

$$x: \ -2x + 3y = 0 \Rightarrow y = 2x/3,$$
$$y: \ 7x + 6y = 0 \Rightarrow y = -7x/6.$$

We now plot the velocity field, nullclines and several trajectories:

```
> example9.flowField <- flowField(example9, x.lim = c(-3,
+ 3), y.lim = c(-3, 3), points = 19, add = FALSE, xlab =
+ "x", ylab = "y")
> grid()
> example9.nullclines <- nullclines(example9, x.lim = c(-3,
+ 3), y.lim = c(-3, 3))
> y0 <- matrix(c(1, 0, -3, 2, 2, -2, -2, -2), ncol = 2,
+ nrow = 4, byrow = TRUE)
> example9.trajectory <- trajectory(example9, y0 = y0,
+ t.end = 10, xlab = "x", ylab = "y")
```



Now from the equations of the nullclines or from the plot we can see that the (only) equilibrium point is at (0,0), and that it appears to be a saddle point. To confirm this we use the Jacobian:

$$\mathbf{J} = \begin{pmatrix} -2 & 3 \\ 7 & 6 \end{pmatrix},$$
$$\implies T = 4, \Delta = -33, T^2 - 4\Delta = 148.$$

Thus (0,0) is a saddle. We can also perform this analysis using `stability`:

```
> example9.stability <- stability(example9, y.star = c(0,
+ 0))
```

```
T: 4    Delta: -33    Discriminant: 148    Classification:
Saddle
```

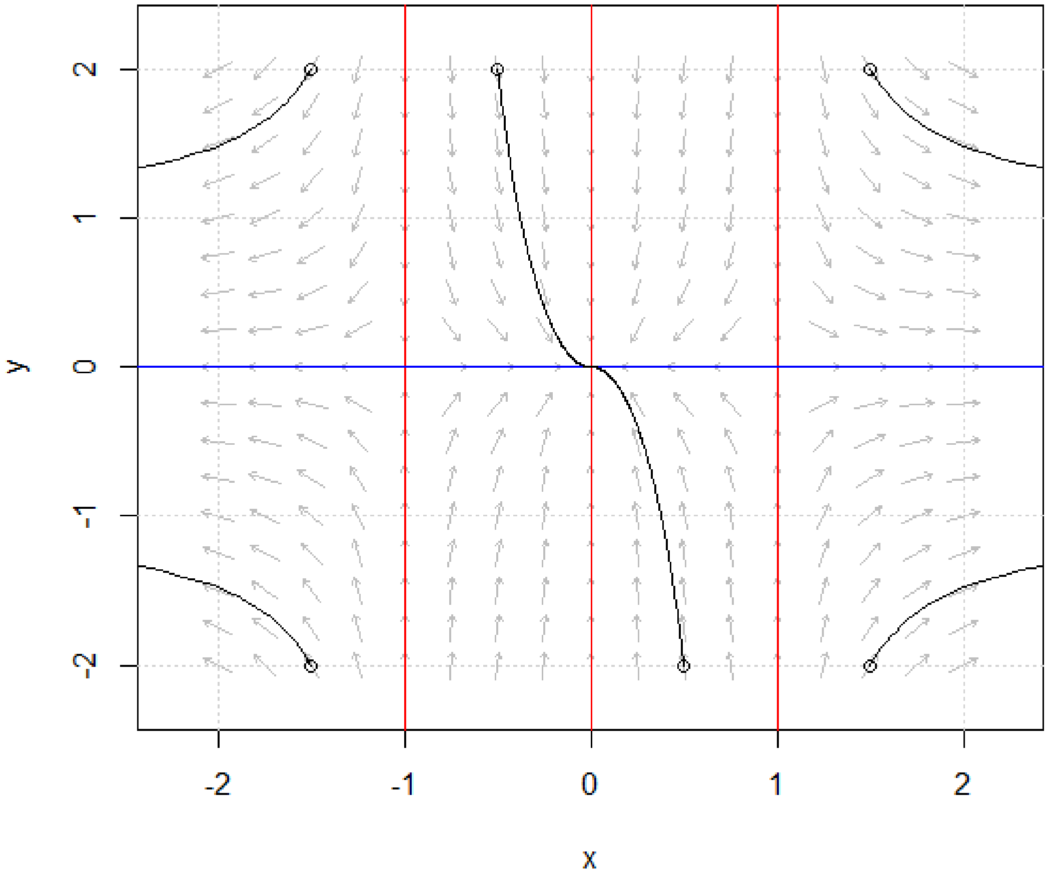
Exercise 8: We now proceed through each of these example non-linear two dimensional autonomous ODE systems:

a) We begin by identifying the nullclines by setting the derivatives to zero:

$$x: \; -x^3 + x = 0 \Longrightarrow x(x-1)(x+1) = 0 \Longrightarrow x = 0,\ 1,\ -1.$$
$$y: \; -2y = 0 \Longrightarrow y = 0.$$

We now plot the velocity field, nullclines and several trajectories:

```
> example10.flowField <- flowField(example10, x.lim = c(-2,
+ 2), y.lim = c(-2, 2), points = 17, add = FALSE, xlab =
+ "x", ylab = "y")
> grid()
> example10.nullclines <- nullclines(example10, x.lim =
+ c(-3, 3), y.lim = c(-3, 3), points = 200)
> y0 <- matrix(c(1.5, 2, -0.5, 2, 0.5, -2, -1.5, 2, 1.5, -2,
+ -1.5, -2), ncol = 2, nrow = 6, byrow = TRUE)
> example10.trajectory <- trajectory(example10, y0 = y0,
+ t.end = 10, xlab = "x", ylab = "y")
```



Now from the equations of the nullclines (and from the plot) we can see that the equilibrium points are at (0,0), (1,0) and (−1,0). To classify them we use the Jacobian:

$$\mathbf{J} = \begin{pmatrix} -1 + 3x_*^2 & 0 \\ 0 & -2 \end{pmatrix}.$$

Thus we have:

Equilibrium Point	$\boldsymbol{\Delta}$	$\boldsymbol{T^2 - 4\Delta}$	$\boldsymbol{T}$	**Classification**
(0,0)	2	1	−3	Stable node
(1,0)	−4	16	0	Saddle
(−1,0)	−4	16	0	Saddle

Finally, we confirm this using `stability`:

```
> example10.stability.1 <- stability(example10, y.star =
+ c(0, 0))
T: -3   Delta: 2   Discriminant: 1   Classification: Stable
node
> example10.stability.2 <- stability(example10, y.star =
+ c(1, 0), h = 1e-8)
T: 1.004952e-08   Delta: -4   Discriminant: 16
Classification: Saddle
> example10.stability.3 <- stability(example10, y.star =
+ c(-1, 0), h = 1e-8)
T: -2.325717e-08   Delta: -4   Discriminant: 16
Classification: Saddle
```

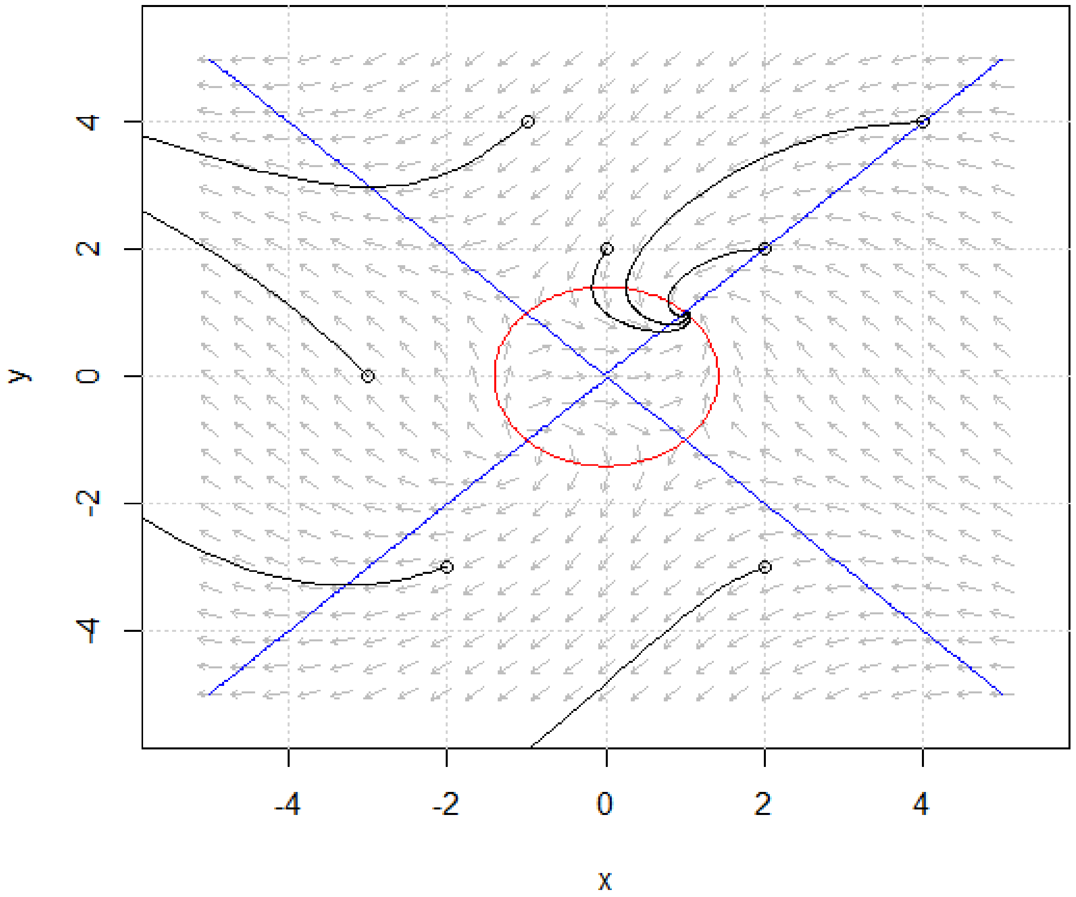
Note that this example provides the first case we've come across where stability is unable to precisely determine the Jacobian. This can sometimes be the case for particularly tricky non-linear systems.

b) We begin by identifying the nullclines by setting the derivatives to zero:

$$x : \; 2 - x^2 - y^2 = 0 \Longrightarrow x^2 + y^2 = 2.$$
$$y : x^2 - y^2 = 0 \Longrightarrow y = \pm x.$$

We now plot the velocity field, nullclines and several trajectories:

```
> example13.flowField <- flowField(example13, x.lim = c(-5,
+ 5), y.lim = c(-5, 5), points = 25, add = FALSE, xlab =
+ "x", ylab = "y")
> grid()
> example13.nullclines <- nullclines(example13, x.lim =
+ c(-5, 5), y.lim = c(-5, 5), points = 200)
> y0 <- matrix(c(2, 2, -3, 0, 0, 2, -1, 4, 4, 4, -2, -3, 2,
+ -3), ncol = 2, nrow = 7, byrow = TRUE)
> example13.trajectory <- trajectory(example13, y0 = y0,
+ t.end = 10, xlab = "x", ylab = "y")
```



Now from the equations of the nullclines (and from the plot) we can see that the equilibrium points are at $(\sqrt{2},\sqrt{2})$, $(\sqrt{2},-\sqrt{2})$, $(-\sqrt{2},\sqrt{2})$ and $(-\sqrt{2},-\sqrt{2})$. To classify them we use the Jacobian:

$$\mathbf{J} = \begin{pmatrix} -2x_* & -2y_* \\ 2x_* & -2y_* \end{pmatrix}.$$

Thus we have:

Equilibrium Point	$\boldsymbol{\Delta}$	$\boldsymbol{T^2 - 4\Delta}$	$\boldsymbol{T}$	**Classification**
$(\sqrt{2},\sqrt{2})$	16	-32	$-4\sqrt{2}$	Stable focus
$(\sqrt{2},\sqrt{2})$	-16	64	0	Saddle
$(\sqrt{2},\sqrt{2})$	-16	64	0	Saddle
$(\sqrt{2},\sqrt{2})$	16	-32	$4\sqrt{2}$	Unstable focus

Finally, we confirm this using `stability`:

```
> example13.stability.1 <- stability(example13, y.star =
+ c(2*sqrt(2), 2*sqrt(2)))
T: -11.31371   Delta: 64   Discriminant: -128
Classification: Stable focus
> example13.stability.2 <- stability(example13, y.star =
+ c(2*sqrt(2), -2*sqrt(2)))
T: -1.776357e-07   Delta: -64   Discriminant: 256
Classification: Saddle
> example13.stability.3 <- stability(example13, y.star =
+ c(-2*sqrt(2), 2*sqrt(2)))
T: -1.776357e-07   Delta: -64   Discriminant: 256
Classification: Saddle
> example13.stability.4 <- stability(example13, y.star =
+ c(-2*sqrt(2), -2*sqrt(2)))
T: 11.31371   Delta: 64   Discriminant: -128
Classification: Unstable focus
```

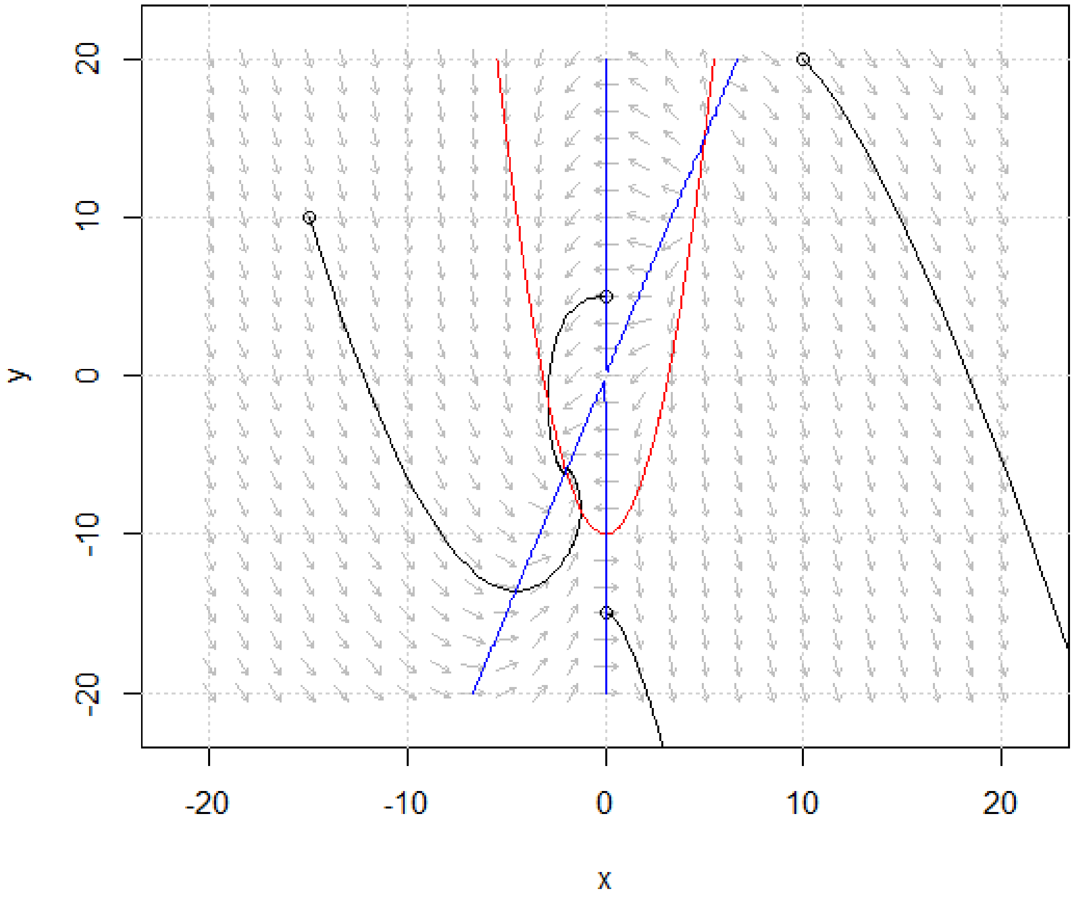
Again, we see in some instances here that stability was unable to perfectly determine the value of the trace.

c) We begin by identifying the nullclines by setting the derivatives to zero:

$$x : x^2 - y - 10 = 0 \Longrightarrow y = x^2 - 10.$$
$$y : -3x^2 + xy = 0 \Longrightarrow x(-3x + y) = 0 \Longrightarrow x = 0 \text{ or } y = 3x.$$

We now plot the velocity field, nullclines and several trajectories:

```
> example14.flowField <- flowField(example14, x.lim = c(-20,
+ 20), y.lim = c(-20, 20), points = 25, add = FALSE, xlab =
+ "x", ylab = "y")
> grid()
> example14.nullclines <- nullclines(example14, x.lim = c(-
20, 20), y.lim = c(-20, 20), points = 200)
> y0 <- matrix(c(-15, 10, 10, 20, 0, 5, 0, -15), ncol = 2,
+ nrow = 4, byrow = TRUE)
> example14.trajectory <- trajectory(example14, y0 = y0,
+ t.end = 10, xlab = "x", ylab = "y")
```



Now from the equations of the nullclines, substituting one into the other, we can see that the equilibrium points are at $(0, -10)$, $(5,15)$ and $(-2, -6)$. To classify them we use the Jacobian:

$$\mathbf{J} = \begin{pmatrix} 2x_* & -1 \\ -6x_* + y_* & x_* \end{pmatrix}.$$

Thus we have:

Equilibrium Point	Δ	$T^2 - 4\Delta$	T	Classification
$(0, -10)$	-10	40	0	Saddle
$(5,15)$	35	85	15	Unstable node
$(-2, -6)$	14	-20	-6	Stable focus

Finally, we confirm this using `stability`:

```
> example14.stability.1 <- stability(example14, y.star =
+ c(0, -10))
T: 1.065814e-07   Delta: -10   Discriminant: 40
Classification: Saddle
> example14.stability.2 <- stability(example14, y.star =
+ c(5, 15))
```

```
T: 15   Delta: 35   Discriminant: 85   Classification:
Unstable node
> example14.stability.3 <- stability(example14, y.star =
+ c(-2, -6))
T: -6   Delta: 14   Discriminant: -20   Classification:
Stable focus
```

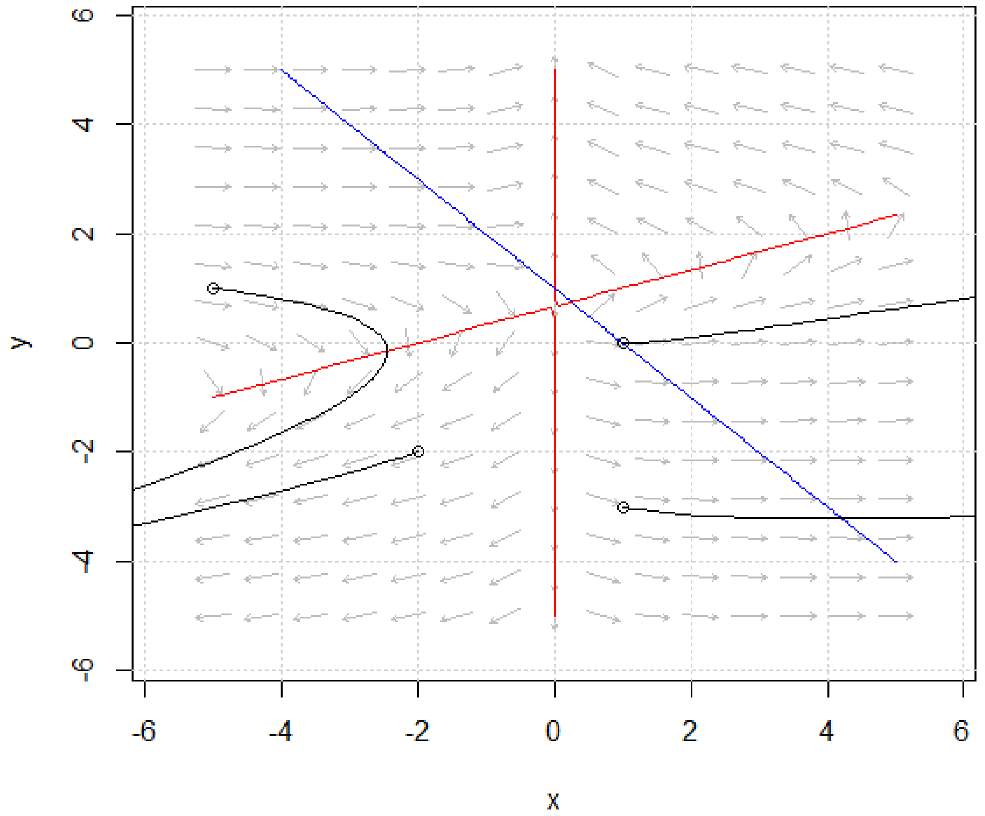
d) We begin by identifying the nullclines by setting the derivatives to zero:

$$x:\ x^2 - 3xy + 2x = 0 \Longrightarrow x(x - 3y + 2) = 0 \Longrightarrow x = 0 \text{ or } y = \frac{x+2}{3}.$$

$$y: x + y - 1 = 0 \Longrightarrow y = 1 - x.$$

We now plot the velocity field, nullclines and several trajectories:

```
> example15.flowField <- flowField(example15, x.lim = c(-5,
+ 5), y.lim = c(-5, 5), points = 15, add = FALSE, xlab =
+ "x", ylab = "y")
> grid()
> example15.nullclines <- nullclines(example15, x.lim =
+ c(-5, 5), y.lim = c(-5, 5), points = 100)
> y0 <- matrix(c(-5, 1, 1, 0, 1, -3, -2, -2), ncol = 2,
+ nrow = 4, byrow = TRUE)
> example15.trajectory <- trajectory(example15, y0 = y0,
+ t.end = 10, xlab = "x", ylab = "y")
```



Now from the equations of the nullclines, substituting one into the other, we can see that the equilibrium points are at (0,1), (1/4,3/4). To classify them we use the Jacobian:

$$\mathbf{J} = \begin{pmatrix} 2x_* - 3y_* + 2 & -3x_* \\ 1 & 1 \end{pmatrix}.$$

Thus we have:

Equilibrium Point	$\boldsymbol{\Delta}$	$\boldsymbol{T^2 - 4\Delta}$	$\boldsymbol{T}$	**Classification**
(0,1)	-1	4	0	Saddle
(1/4,3/4)	1	-2.4375	1.25	Unstable focus

Finally, we confirm this using `stability`:

```
> example15.stability.1 <- stability(example15, y.star =
+ c(0, 1))
T: 3.922529e-09   Delta: -1   Discriminant: 4
Classification: Saddle
> example15.stability.2 <- stability(example15, y.star =
+ c(0.25, + 0.75))
T: 1.25   Delta: 1   Discriminant: -2.4375   Classification:
Unstable focus
```

Exercise 9: We first produce a derivative function for this exercise:

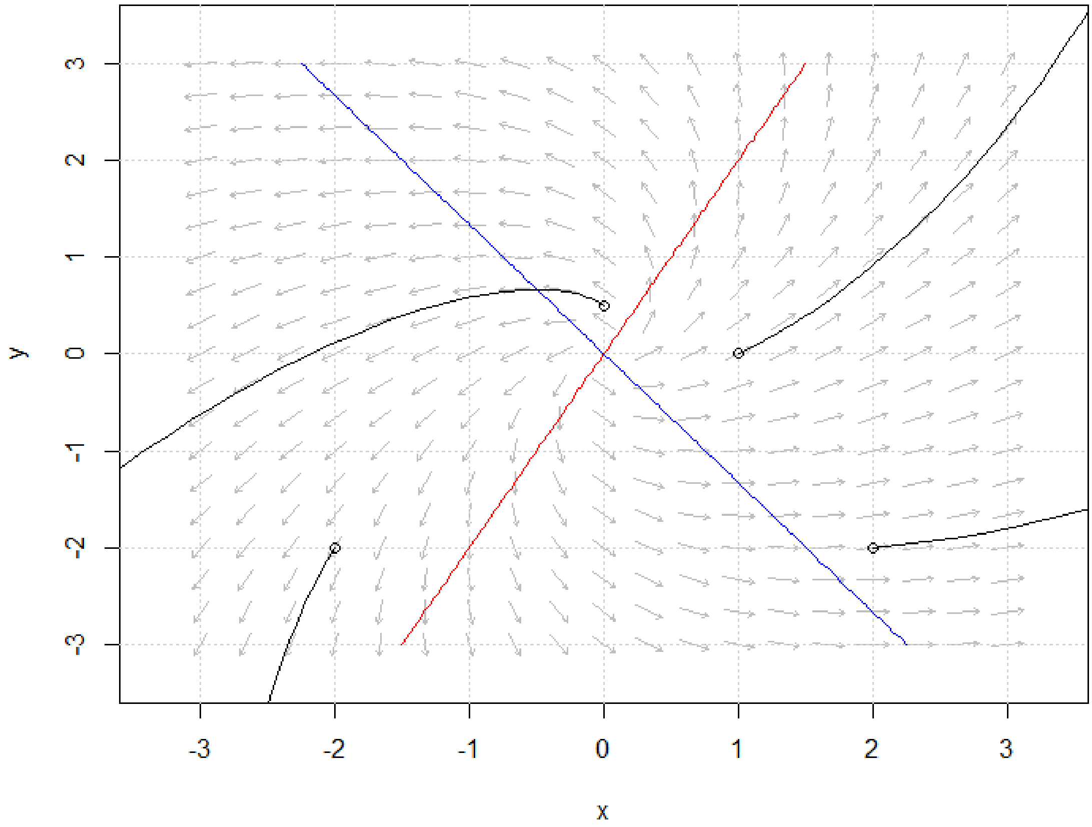
```
exercise9 <- function(t, y, parameters){
  x <- y[1]
  y <- y[2]
  dy <- rep(0, 2)
  dy[1] <- 6*x - 3*y
  dy[2] <- 4*x + 3*y
  list(dy)
}
```

Next, we identify the nullclines by setting the derivatives to zero:

$$x: \ 6x - 3y = 0 \Longrightarrow y = 2x,$$
$$y: \ 4x + 3y = 0 \Longrightarrow y = -4x/3.$$

We now plot the velocity field, nullclines and several trajectories:

```
> exercise9.flowField <- flowField(exercise9, x.lim = c(-3,
+ 3), y.lim = c(-3, 3), points = 19, add = FALSE, xlab =
+ "x", ylab = "y")
> grid()
> exercise9.nullclines <- nullclines(exercise9, x.lim =
+ c(-3, 3), y.lim = c(-3, 3))
> y0 <- matrix(c(1, 0, 0, 0.5, 2, -2, -2, -2), ncol = 2,
+ nrow = 4, byrow = TRUE)
> exercise9.trajectory <- trajectory(exercise9, y0 = y0,
+ t.end = 10, xlab = "x", ylab = "y")
```



Now from the equations of the nullclines or from the plot we can see that the (only) equilibrium point is at (0,0), and that it appears to be an unstable focus. To confirm this we use the Jacobian:

$$\mathbf{J} = \begin{pmatrix} 6 & -3 \\ 4 & 3 \end{pmatrix},$$

$$\Longrightarrow T = 9, \Delta = -33, T^2 - 4\Delta = 148.$$

Thus (0,0) is a saddle. We can also perform this analysis using `stability`:

```
> exercise9.stability <- stability(exercise9, y.star = c(0,
+ 0))
T: 9   Delta: 30   Discriminant: -39   Classification:
Unstable focus
```

Exercise 10: We first produce a derivative function for this exercise:

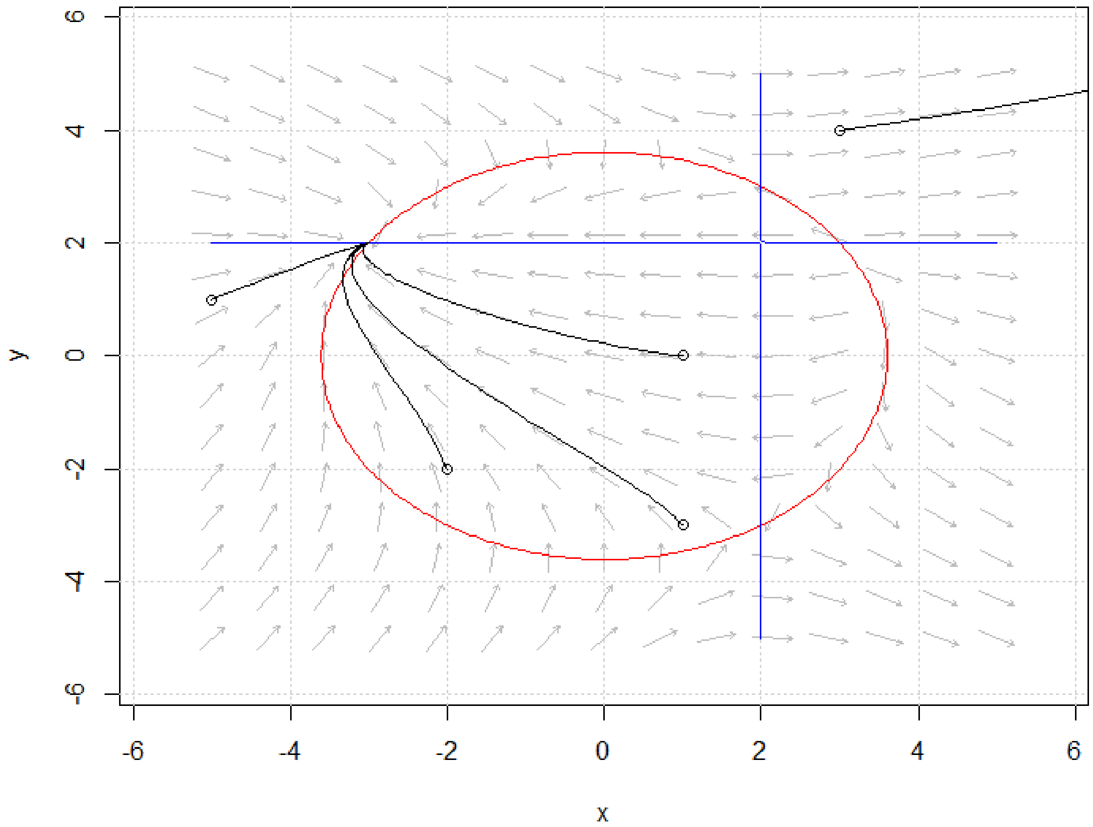
```
exercise10 <- function (t, y, parameters){
  x <- y[1]
  y <- y[2]
  dy <- rep(0, 2)
  dy[1] <- x^2 + y^2 - 13
  dy[2] <- x*y - 2*x - 2*y + 4
  list(dy)
}
```

Next, we identify the nullclines by setting the derivatives to zero:

$$x:\ x^2 + y^2 - 13 = 0 \Longrightarrow x^2 + y^2 = 13.$$
$$y: xy - 2x - 2y + 4 = 0 \Longrightarrow (x-2)(y-2) = 0 \Longrightarrow x = 2 \text{ or } y = 2.$$

We now plot the velocity field, nullclines and several trajectories:

```
> exercise10.flowField <- flowField(exercise10, x.lim = c(-
5, + 5), y.lim = c(-5, 5), points = 15, add = FALSE, xlab =
+ "x", ylab = "y")
> grid()
> exercise10.nullclines <- nullclines(exercise10, x.lim =
+ c(-5, 5), y.lim = c(-5, 5), points = 100)
> y0 <- matrix(c(-5, 1, 1, 0, 1, -3, -2, -2, 3, 4), ncol =
+ 2, nrow = 5, byrow = TRUE)
> exercise10.trajectory <- trajectory(exercise10, y0 = y0,
+ t.end = 10, xlab = "x", ylab = "y")
```



Now from the equations of the nullclines, substituting one into the other, we can see that the equilibrium points are at (2,3), (2, −3), (3,2) and (−3,2). To classify them we use the Jacobian:

$$\mathbf{J} = \begin{pmatrix} 2x_* & 2y_* \\ y_* - 2 & x_* - 2 \end{pmatrix}.$$

Thus we have:

Equilibrium Point	$\boldsymbol{\Delta}$	$\boldsymbol{T^2 - 4\Delta}$	$\boldsymbol{T}$	**Classification**
(2,3)	-6	40	4	Saddle
(2, −3)	-30	136	4	Saddle
(3,2)	6	25	7	Unstable node
(−3,2)	30	1	-11	Stable node

Finally, we confirm this using `stability`:

```
> exercise10.stability.1 <- stability(exercise10, y.star =
+ c(2, 3))
T: 4    Delta: -6    Discriminant: 40    Classification: Saddle
> exercise10.stability.2 <- stability(exercise10, y.star =
+ c(2, -3))
```

```
T: 4   Delta: -30   Discriminant: 136   Classification:
Saddle
> exercise10.stability.3 <- stability(exercise10, y.star =
+ c(3, 2))
T: 7   Delta: 6   Discriminant: 25   Classification:
Unstable node
> exercise10.stability.4 <- stability(exercise10, y.star =
+ c(-3, 2))
T: -11   Delta: 30   Discriminant: 0.9999998
Classification: Stable node
```

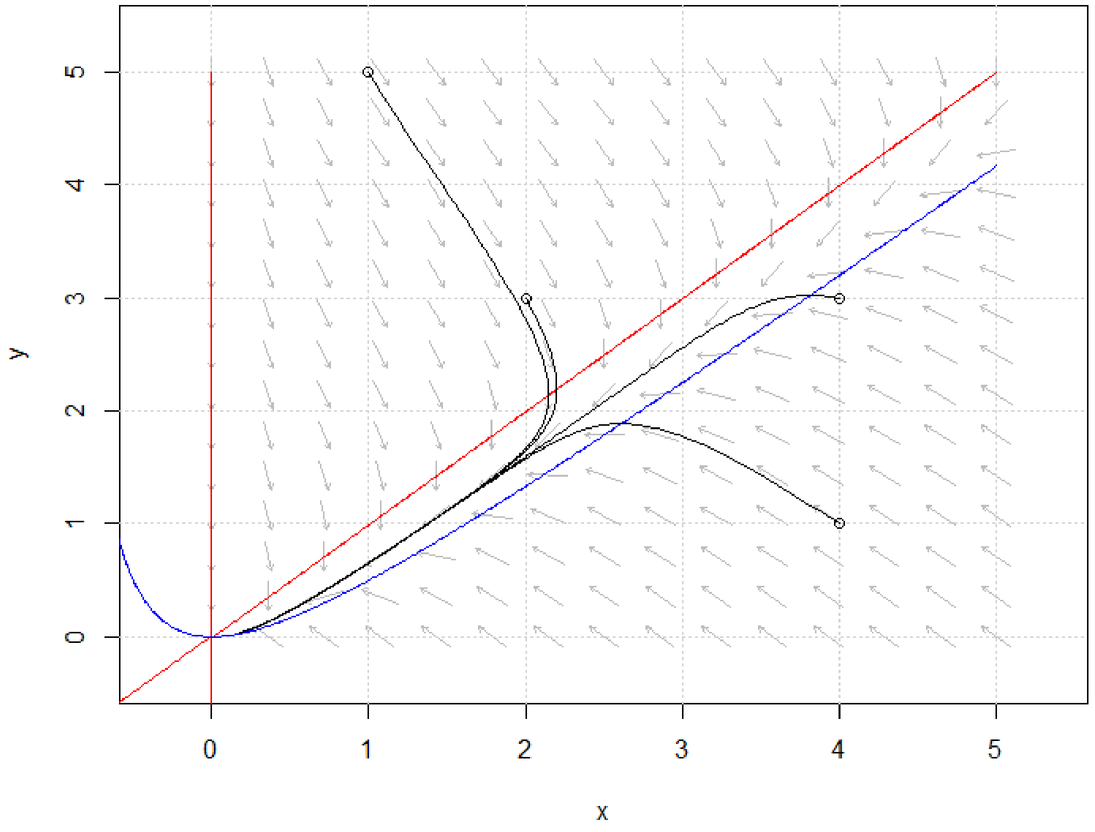
Exercise 11: Here, start by identifying the nullclines by setting the derivatives to zero ($\alpha \neq 0$ so there's no problem dividing through):

$$x : -x^2 + \alpha xy = 0 \Longrightarrow x(-x + \alpha y) = 0 \Longrightarrow x = 0 \text{ or } y = \frac{x}{\alpha},$$

$$y : x^2 - \alpha xy - y = 0 \Longrightarrow y = \frac{x^2}{\alpha x + 1}.$$

We now plot the velocity field, nullclines and add several trajectories:

```
> lindemannMechanism.flowField <-
+ flowField(lindemannMechanism, x.lim = c(0, 5), y.lim =
+ c(0, 5), parameters = 1, points = 15, add = FALSE, xlab =
+ "x", ylab = "y")
> grid()
> lindemannMechanism.nullclines <-
+ nullclines(lindemannMechanism, x.lim = c(-1, 5), y.lim =
+ c(-1, 5), parameters = 1, points = 500)
> y0 <- matrix(c(1, 5, 4, 3, 2, 3, 4, 1), ncol = 2, nrow =
+ 4, byrow = TRUE)
> lindemannMechanism.trajectory <-
+ trajectory(lindemannMechanism, y0 = y0, t.end = 10,
+ parameters = 1, xlab = "x", ylab = "y")
```



Now from the equations of the nullclines, we can see that the only equilibrium point is at (0,0). To classify it we use the Jacobian:

$$\mathbf{J} = \begin{pmatrix} -2x_* + \alpha y & \alpha x_* \\ 2x_* - \alpha y_* & -1 \end{pmatrix}.$$

Thus for (0,0) we have $\Delta = 0$ and we are unable to determine stability. However, referring back to our earlier plot we can indeed see that (0,0) is stable. Again, this shows how plots can be useful when the Jacobian approach fails.

Exercise 12: First, note that the restriction to a population size less than 10 simply means that $x + y \leq 10$ at all times. Next, identify the nullclines by setting the derivatives to zero:

$$\begin{aligned} x &: \ -\beta xy = 0 \Longrightarrow x = 0 \text{ or } y = 0, \\ y &: \beta xy - \nu y = 0 \Longrightarrow y(\beta x - \nu) = 0 \Longrightarrow y = 0 \text{ or } x = \nu/\beta. \end{aligned}$$

So we now have two cases, based upon whether the nullcline $x = \nu/\beta$ takes a value less than 10 or not. Before we plot these two cases, we will identify the equilibrium points and classify them.

From the equations of the nullclines, we can see that we have a line of equilibrium points given by $y = 0$. To classify them we turn to the Jacobian:

$$\mathbf{J} = \begin{pmatrix} -\beta y_* & -\beta x_* \\ \beta y_* & \beta x_* - \nu \end{pmatrix}.$$

So if $y_* = 0$ we have:

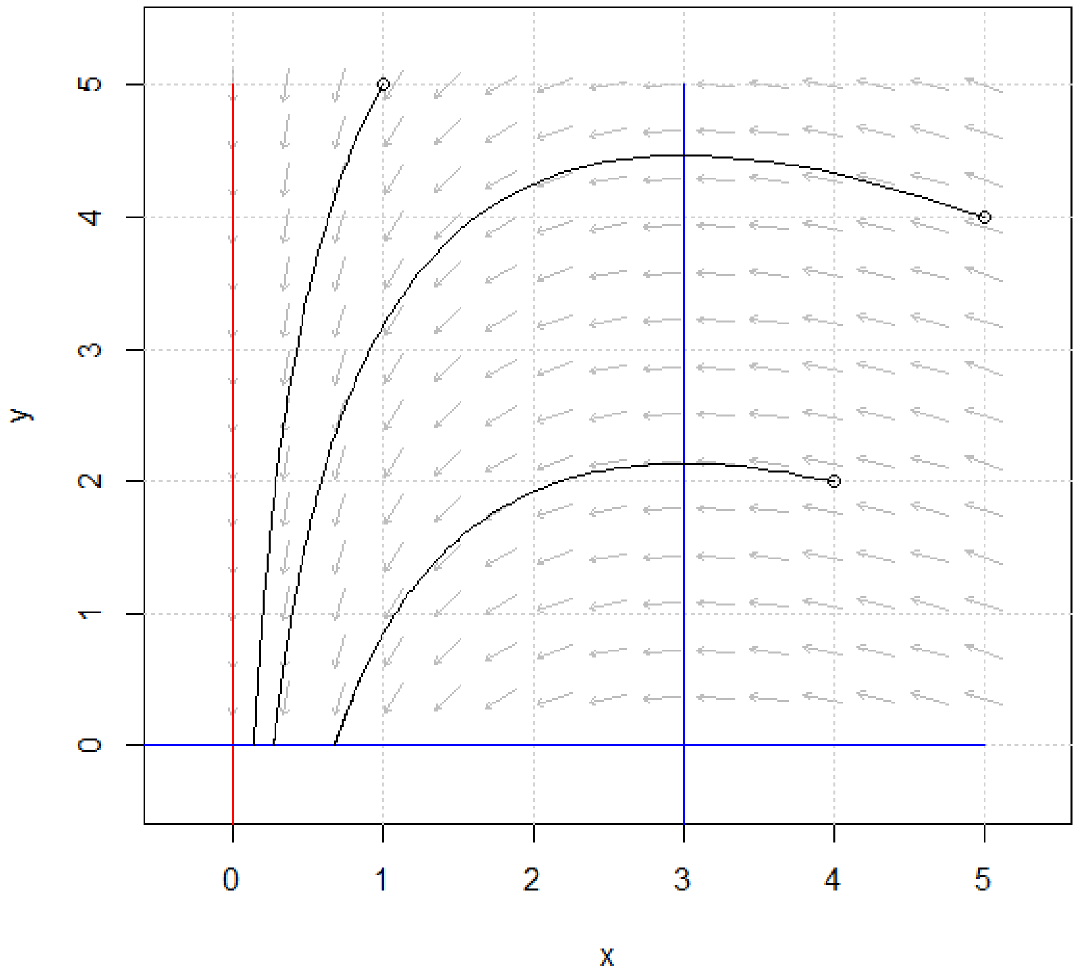
$$\mathbf{J} = \begin{pmatrix} 0 & -\beta x_* \\ 0 & \beta x_* - \nu \end{pmatrix},$$

and $\Delta = 0$, meaning we cannot determine stability using this method. We will have to use our following plots instead. We can confirm that the Taylor approach fails using `stability` (taking the example $\beta = 1, \nu = 3$):

```
> SIR.stability <- stability(SIR, y.star = c(0, 0),
+ parameters = c(1, 3))
T: -3   Delta: 0   Discriminant: 9   Classification:
Indeterminate
```

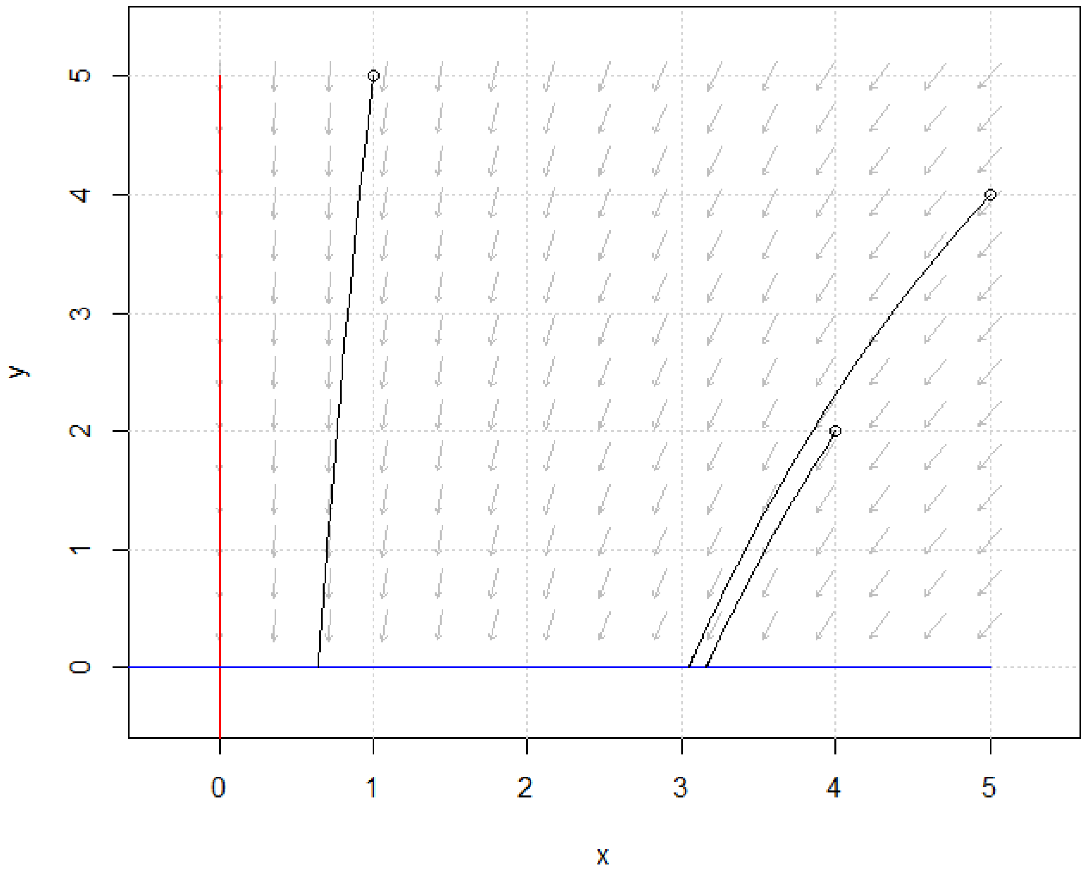
So, let us plot first the case where $\beta = 1, \nu = 3$, so that the nullcline does lie in the admissible region of the plane:

```
> SIR.flowField <- flowField(SIR, x.lim = c(0, 5), y.lim = +
c(0, 5), parameters = c(1, 3), points = 15, add = FALSE, +
xlab = "x", ylab = "y")
> grid()
> SIR.nullclines <- nullclines(SIR, x.lim = c(-1, 5),
+ y.lim = c(-1, 5), parameters = c(1, 3), points = 500)
> y0 <- matrix(c(4, 2, 5, 4, 1, 5), ncol = 2, nrow = 3,
+ byrow = TRUE)
> SIR.trajectory <- trajectory(SIR, y0 = y0, t.end = 10,
+ parameters = c(1, 3), xlab = "x", ylab = "y")
```



Thus, it appears here that the equilibrium points given by $y = 0$ will be stable. Now however, we plot the velocity field, nullclines and trajectories in the case where $\beta = 1$ and $\nu = 12$, such that the nullcline $x = \nu/\beta$ does not lie in the admissible region of the plane:

```
> SIR.flowField <- flowField(SIR, x.lim = c(0, 5), y.lim = +
c(0, 5), parameters = c(1, 12), points = 15, add = FALSE, +
xlab = "x", ylab = "y")
> grid()
> SIR.nullclines <- nullclines(SIR, x.lim = c(-1, 5),
+ y.lim = c(-1, 5), parameters = c(1, 12), points = 500)
> y0 <- matrix(c(4, 2, 5, 4, 1, 5), ncol = 2, nrow = 3,
+ byrow = TRUE)
> SIR.trajectory <- trajectory(SIR, y0 = y0, t.end = 10,
+ parameters = c(1, 12), xlab = "x", ylab = "y")
```



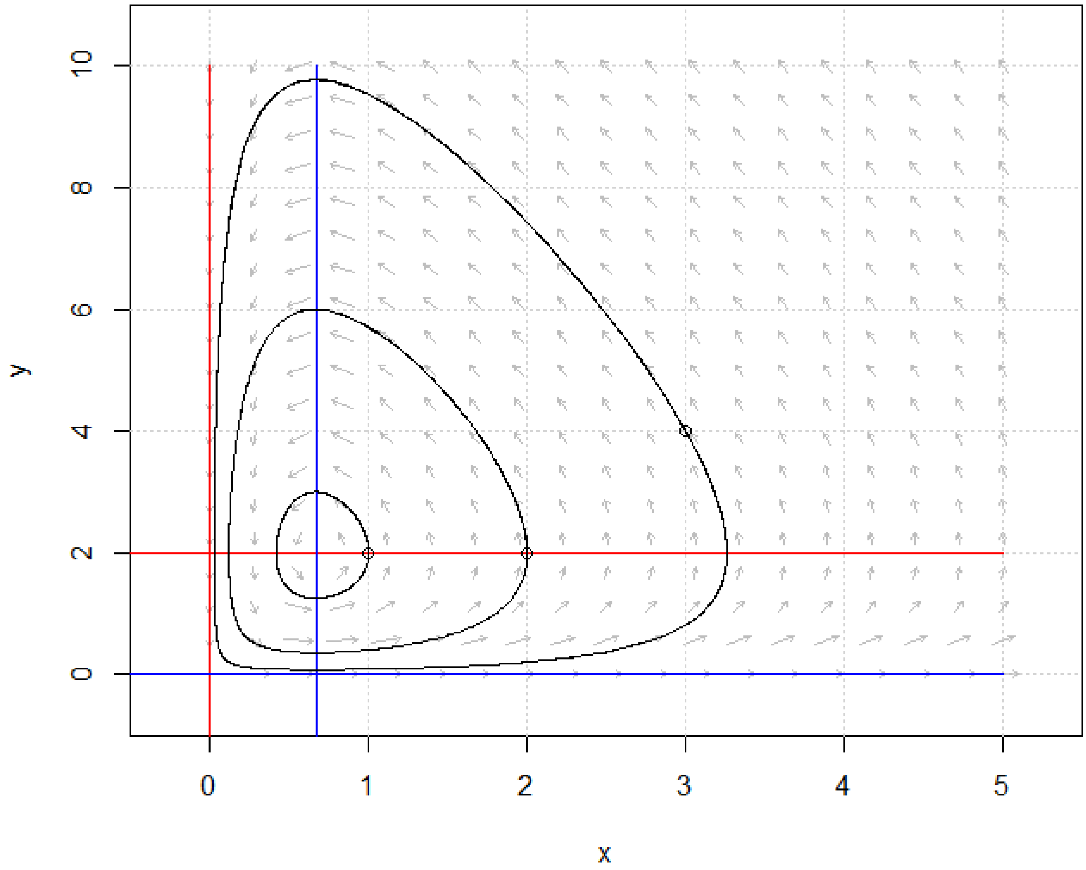
Again it appears that the equilibrium points will be stable. Thus, biologically, our analysis here clearly means that regardless of the initial condition, we eventually approach a disease free equilibrium. However, when $\nu/\beta < 10$, the value of y initially increases. This is indicative of an epidemic. Indeed, this condition on β and ν can be generalised to establish when an epidemic is possible.

Exercise 13: We begin by identifying the nullclines:

$$x : \lambda x - \epsilon xy = 0 \Longrightarrow x(\lambda - \epsilon y) = 0 \Longrightarrow x = 0 \text{ or } y = \lambda/\epsilon,$$
$$y : \eta xy - \delta y = 0 \Longrightarrow y(\eta x - \delta) = 0 \Longrightarrow y = 0 \; or \; x = \delta/\eta.$$

Next, we plot the velocity field, nullclines and several trajectories, for the case $\lambda = 2$, $\epsilon = 1, \eta = 3$ and $\delta = 2$:

```
> lotkaVolterra.flowField <- flowField(lotkaVolterra,
+ x.lim = c(0, 5), y.lim = c(0, 10), parameters = c(2, 1, +
3, 2), points = 19, add = FALSE, xlab = "x", ylab = "y")
> grid()
> lotkaVolterra.nullclines <- nullclines(lotkaVolterra,
+ x.lim = c(-1, 5), y.lim = c(-1, 10), parameters = c(2, 1,
+ 3, 2), points = 500)
> y0 <- matrix(c(1, 2, 2, 2, 3, 4), ncol = 2, nrow = 3,
+ byrow = TRUE)
> lotkaVolterra.trajectory <- trajectory(lotkaVolterra, y0 =
+ y0, t.end = 10, parameters = c(2, 1, 3, 2), xlab = "x",
+ ylab = "y")
```



Now from the equations of the nullclines, we can see that there are two equilibrium points; at (0,0) and $(\delta/\eta, \lambda/\epsilon)$. To classify them we use the Jacobian:

$$\mathbf{J} = \begin{pmatrix} \lambda - \epsilon y_* & -\epsilon x_* \\ \eta y_* & \eta x_* - \delta \end{pmatrix}.$$

Thus we have:

Equilibrium Point	$\boldsymbol{\Delta}$	$\boldsymbol{T^2 - 4\Delta}$	$\boldsymbol{T}$	**Classification**
(0,0)	$-\delta\lambda < 0$	N/A	N/A	Saddle
$(\delta/\eta, \lambda/\epsilon)$	$\delta\lambda > 0$	$-4\delta\lambda < 0$	0	Centre

Indeed, looking back at our earlier plot we can see this is the case.

Finally, we verify this result for our specific case using `stability`:

```
> lotkaVolterra.stability.1 <- stability(lotkaVolterra,
+ y.star = c(0, 0), parameters = c(2, 1, 3, 2))
T: 0   Delta: -4   Discriminant: 16   Classification: Saddle
> lotkaVolterra.stability.2 <- stability(lotkaVolterra,
+ y.star = c(2/3, 2), parameters = c(2, 1, 3, 2))
T: 0   Delta: 4   Discriminant: -16   Classification: Centre
```

Biologically, this simply means that no matter what the initial starting values, according to this model the numbers of predator and prey will oscillate.

Exercise 14: We begin by identifying the x-nullclines:

$$r_1 x\left(\frac{K_1 - x - \alpha_{12}y}{K_1}\right) = 0 \Longrightarrow x = 0 \text{ or } K_1 - x - \alpha_{12}y = 0,$$

$$\Longrightarrow x = 0 \text{ or } y = \frac{K_1 - x}{\alpha_{12}}.$$

And now the y-nullclines:

$$r_2 y\left(\frac{K_2 - y - \alpha_{21}x}{K_2}\right) = 0 \Longrightarrow y = 0 \text{ or } K_2 - y - \alpha_{21}x = 0,$$

$$\Longrightarrow y = 0 \text{ or } y = K_2 - \alpha_{21}x.$$

Here things get slightly tricky; there are four cases based on how the nullclines sit relative to each other. The lines:

$$y = \frac{K_1 - x}{\alpha_{12}} \text{ and } y = K_2 - \alpha_{21}x,$$

can cross either way around, or one can lie completely above the other. To identify the conditions for each case we examine the intercepts for each axis. For example, for the y-nullcline to lie completely below the x-nullcline it must cross both axes first, i.e.:

$$K_2 < \frac{K_1}{\alpha_{12}} \text{ and } \frac{K_2}{\alpha_{21}} < K_1.$$

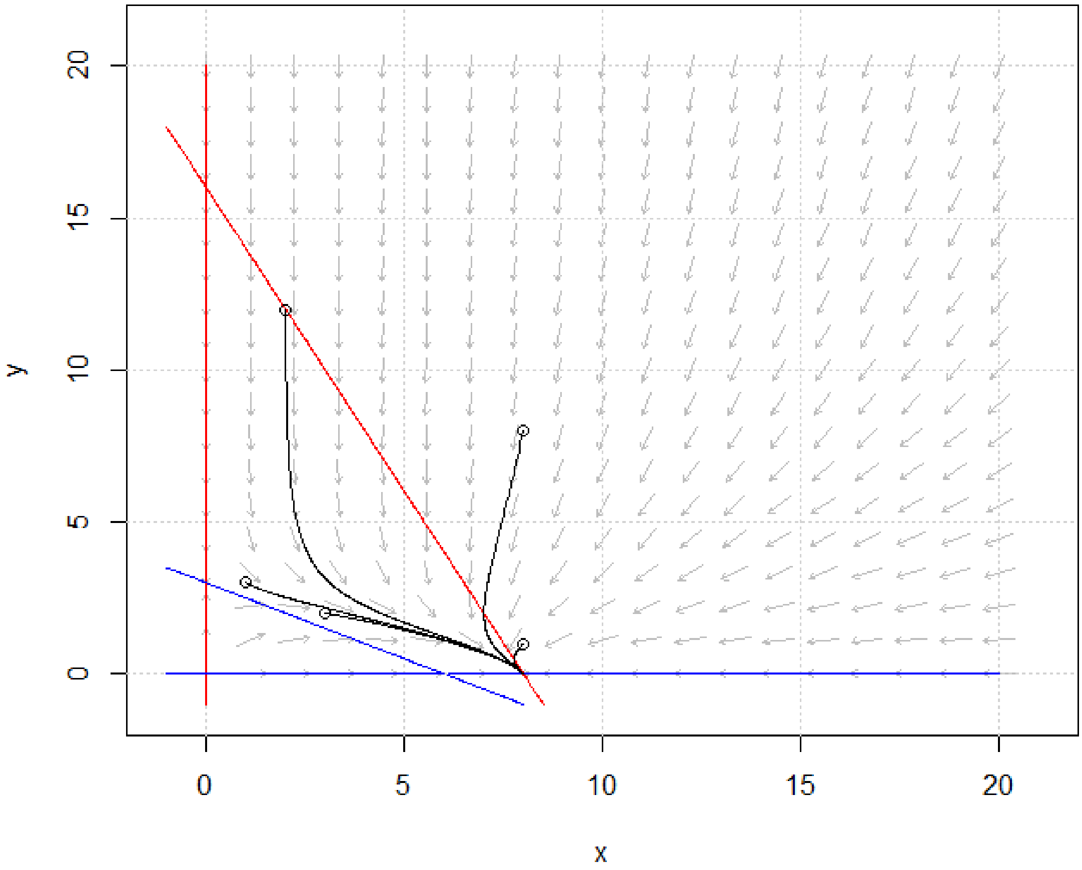
Performing similar calculations for each possibility we can draw up the four cases, and plot the velocity field, nullclines and several trajectories to illustrate it:

Case 1: The y-nullcline lies completely below the x-nullcline (above):

$$K_2 < \frac{K_1}{\alpha_{12}} \text{ and } \frac{K_2}{\alpha_{21}} < K_1.$$

Take $K_1 = 8, K_2 = 3, \alpha_{12} = 0.5, \alpha_{21} = 0.5, r_1 = r_2 = 1$:

```
> competition.1.flowField <- flowField(competition, x.lim =
+ c(0, 20), y.lim = c(0, 20), parameters = c(1, 8, 0.5, 1,
+ 3, 0.5), points = 19, add = FALSE, xlab = "x", ylab = "y")
> grid()
> competition.1.nullclines <- nullclines(competition,
+ x.lim = c(-1, 20), y.lim = c(-1, 20), parameters = c(1, 8,
+ 0.5, 1, 3, 0.5), points = 500)
> y0 <- matrix(c(1, 3, 3, 2, 8, 1, 8, 8, 2, 12), ncol = 2,
+ nrow = 5, byrow = TRUE)
> competition.1.trajectory <- trajectory(competition, y0 =
+ y0,  t.end = 10, parameters = c(1, 8, 0.5, 1, 3, 0.5),
+ xlab = "x", ylab = "y")
```

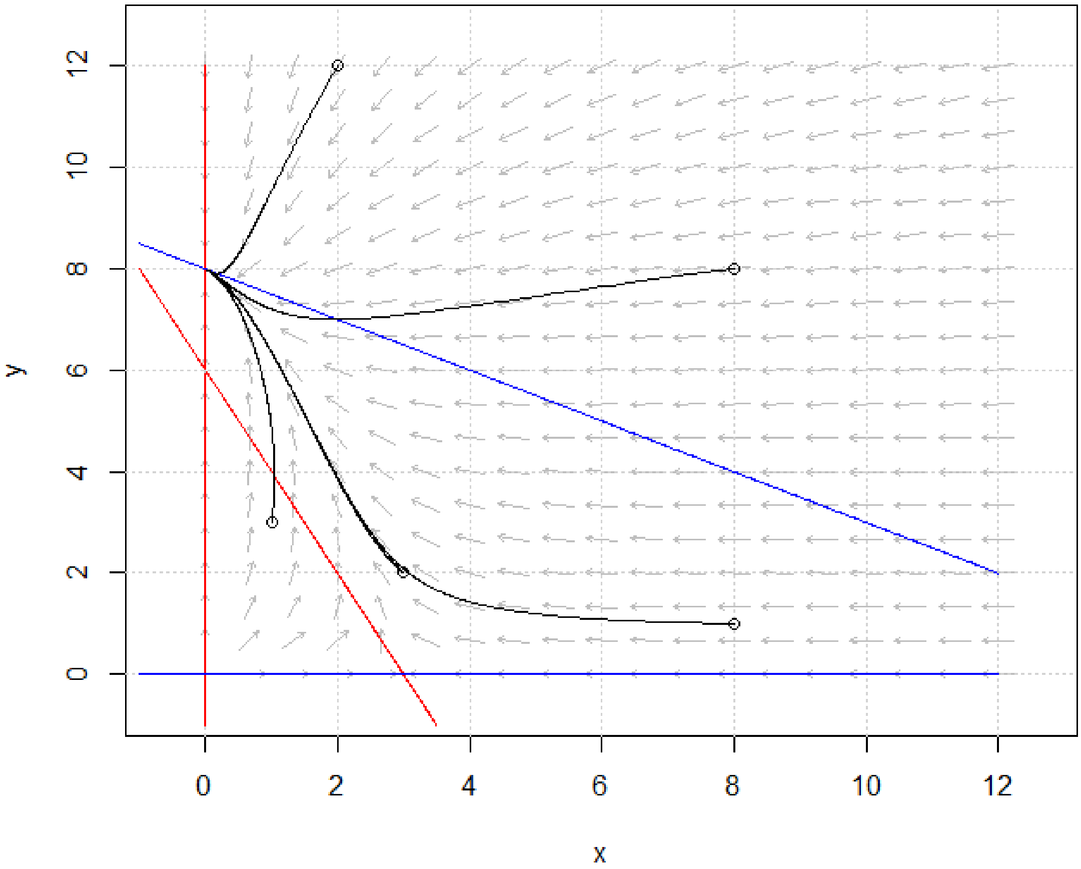


Case 2: The x-nullcline lies completely below the y-nullcline:

$$K_2 > \frac{K_1}{\alpha_{12}} \text{ and } \frac{K_2}{\alpha_{21}} > K_1.$$

Take $K_1 = 3, K_2 = 8, \alpha_{12} = 0.5, \alpha_{21} = 0.75, r_1 = r_2 = 1$:

```
> competition.2.flowField <- flowField(competition, x.lim =
+ c(0, 20), y.lim = c(0, 20), parameters = c(1, 3, 0.5, 1,
+ 8, 0.5), points = 19, add = FALSE, xlab = "x", ylab = "y")
> grid()
> competition.2.nullclines <- nullclines(competition,
+ x.lim = c(-1, 20), y.lim = c(-1, 20), parameters = c(1, 3,
+ 0.5, 1, 8, 0.5), points = 500)
> y0 <- matrix(c(1, 3, 3, 2, 8, 1, 8, 8, 2, 12), ncol = 2,
+ nrow = 5, byrow = TRUE)
> competition.2.trajectory <- trajectory(competition, y0 =
+ y0,  t.end = 10, parameters = c(1, 3, 0.5, 1, 8, 0.5),
+ xlab = "x", ylab = "y")
```

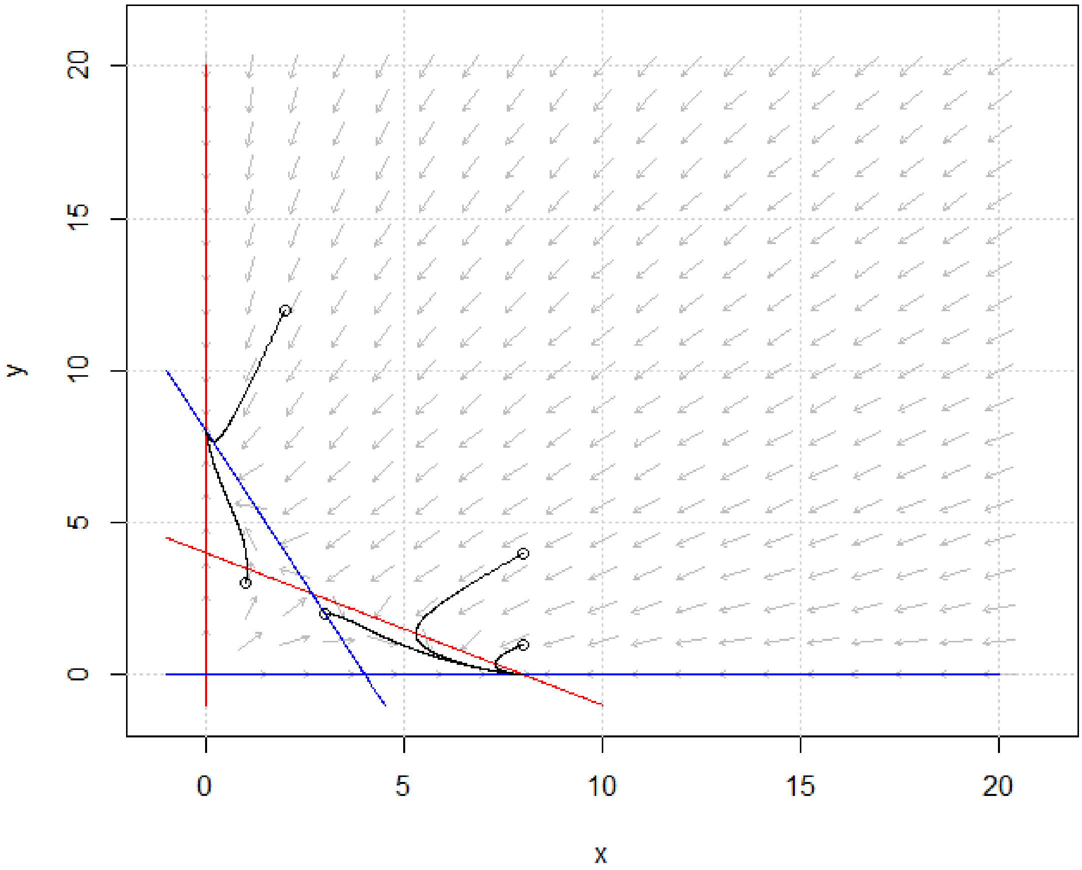


Case 3: The x-nullcline cuts the x-axis first, but the y-axis second:

$$K_2 > \frac{K_1}{\alpha_{12}} \text{ and } \frac{K_2}{\alpha_{21}} < K_1.$$

Take $K_1 = 8, K_2 = 8, \alpha_{12} = 2, \alpha_{21} = 2, r_1 = r_2 = 1$:

```
> competition.3.flowField <- flowField(competition, x.lim =
+ c(0, 20), y.lim = c(0, 20), parameters = c(1, 8, 2, 1,
+ 8, 2), points = 19, add = FALSE, xlab = "x", ylab =
+ "y")
> grid()
> competition.3.nullclines <- nullclines(competition,
+ x.lim = c(-1, 20), y.lim = c(-1, 20), parameters = c(1, 8,
+ 2, 1, 8, 2), points = 500)
> y0 <- matrix(c(1, 3, 3, 2, 8, 1, 8, 4, 2, 12), ncol = 2,
+ nrow = 5, byrow = TRUE)
> competition.3.trajectory <- trajectory(competition, y0 =
+ y0,  t.end = 10, parameters = c(1, 8, 2, 1, 8, 2),
+ xlab = "x", ylab = "y")
```



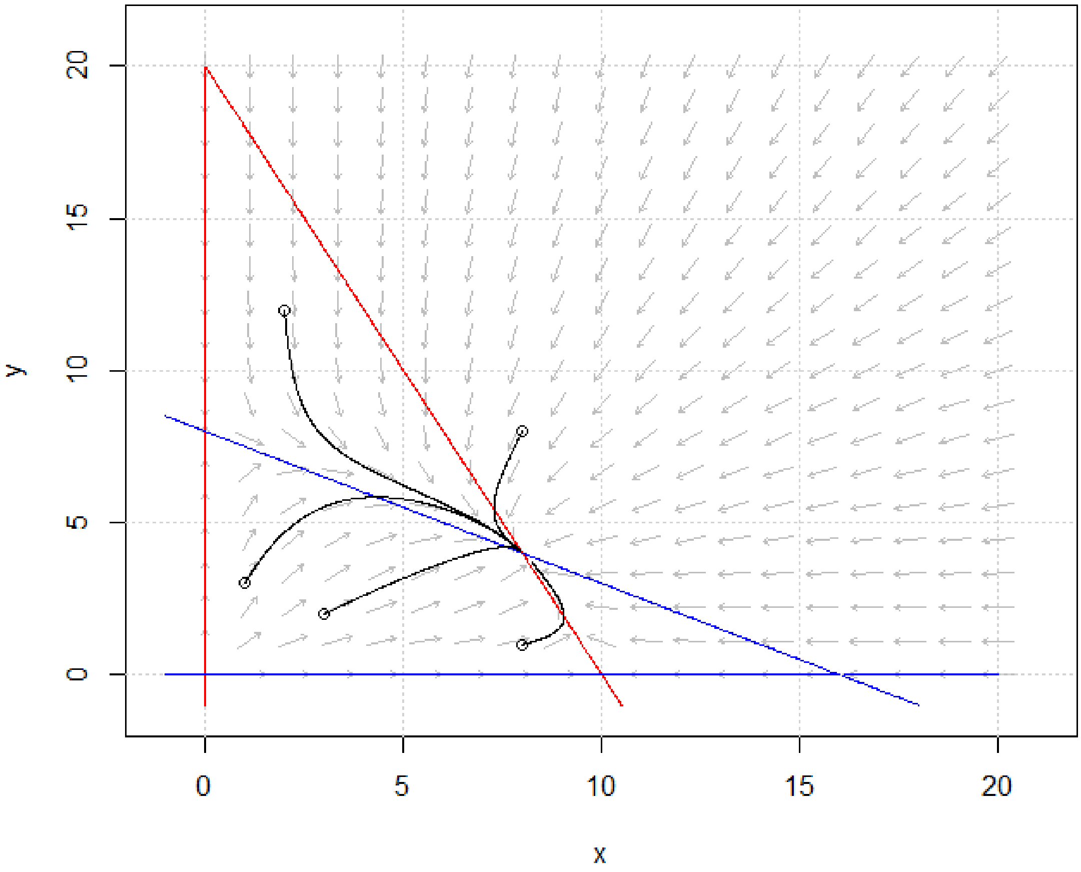
Case 4: The x-nullcline cuts the x-axis second, but the y-axis first:

$$K_2 < \frac{K_1}{\alpha_{12}} \text{ and } \frac{K_2}{\alpha_{21}} > K_1.$$

Take $K_1 = 8, K_2 = 8, \alpha_{12} = 0.5, \alpha_{21} = 0.5, r_1 = r_2 = 1$:

```
> competition.4.flowField <- flowField(competition, x.lim =
+ c(0, 20), y.lim = c(0, 20), parameters = c(1, 8, 0.5, 1,
+ 8, 0.5), points = 19, add = FALSE, xlab = "x", ylab =
+ "y")
```

```
> grid()
> competition.4.nullclines <- nullclines(competition,
+ x.lim = c(-1, 20), y.lim = c(-1, 20), parameters = c(1,
+ 8, 0.5, 1, 8, 0.5), points = 500)
> y0 <- matrix(c(1, 3, 3, 2, 8, 1, 8, 8, 2, 12), ncol = 2,
+ nrow = 5, byrow = TRUE)
> competition.4.trajectory <- trajectory(competition, y0 =
+ y0,  t.end = 10, parameters = c(1, 8, 0.5, 1, 8, 0.5),
+ xlab = "x", ylab = "y")
```



From these plots, and from the nullcline formula, it should be fairly evident that we always have the three equilibrium points given by:

$$(0,0),\ \ (K_1,0),\ \ (0,K_2).$$

However, in cases 3 and 4, we also have a fourth equilibrium point given by the intersection of the two nullclines we studied in detail above. To find the location of this equilibria, we need to substitute one into the other and perform a little bit of algebra. Eventually we find:

$$\frac{1}{1-\alpha_{12}\alpha_{21}}(K_2-\alpha_{21}K_1,\ K_1-\alpha_{12}K_2).$$

Now, in order to determine the stability of these points, we will not perform the messy Jacobian analysis by hand, instead we will refer to our example plots:

- **Case 1:** The system will be driven to the stable point $(K_1, 0)$. The other two equilibria are unstable.
- **Case 2:** The system will be driven to the stable point $(0, K_2)$. The other two equilibria are unstable.
- **Case 3:** Depending upon the initial condition, the system will be driven to one of $(K_1, 0)$ and $(0, K_2)$. The other two equilibria are unstable.
- **Case 4:** The system will be driven to the intersection of the two lines, which we computed on the previous page. The other three equilibria are unstable.

Biologically, this all means that only for case 4 is it possible for both species to co-exist in a stable manner. Else, one species will always drive the other to extinction.