BiomathHW07

Maxwell Greene

April 17, 2020

Problem #1

The following is a generic code to classify the behavior of a linear system based on it's trace and eigenvalues. I will use it in each part of the problem

```
classify <- function(mat)
{
  trace <- sum(diag(mat)); deter <- det(mat)
  delta <- deter - trace^2/4
  if(deter < 0){return("saddle")}
  if(deter ==0)
  {
    if(trace > 0){return("unstable line")}
    if(trace < 0){return("stable line")}
    if(trace ==0){return("uniform motion")}
  }
  if(trace > 0)
  {
    if(delta < 0){return("source")}
    if(delta ==0){return("degenerate source")}
    if(delta > 0){return("spiral source")}
  }
  if(trace ==0){return("center")}
  if(trace < 0)
  {
    if(delta < 0){return("sink")}
    if(delta ==0){return("degenerate sink")}
    if(delta > 0){return("spiral sink")}
  }
}
plotLinearSystem <- function(mat)
{
  simple <- function(t,state,parameters){
    with(as.list(c(state,parameters)),{
    dx <- a*state[1] + b*state[2]
    dy <- c*state[1] + d*state[2]
    list(c(dx,dy))
  })}
  ff  <- flowField(simple,
                   xlim = c(-2, 2), ylim = c(-2, 2),
                   parameters = c(a=mat[1],b=mat[3],c=mat[2],d=mat[4]),
                   points = 11,add = FALSE)
  state <- matrix(c(1,1,1,-1,-1,1,-1,-1,0,2,0,-2,-1,0,1,0),
                                  8, 2, byrow = TRUE)
  trajs <- trajectory(simple,y0 = state, tlim = c(0, 10),
                      parameters = c(a=mat[1],b=mat[3],c=mat[2],d=mat[4]),add=TRUE)
}
```

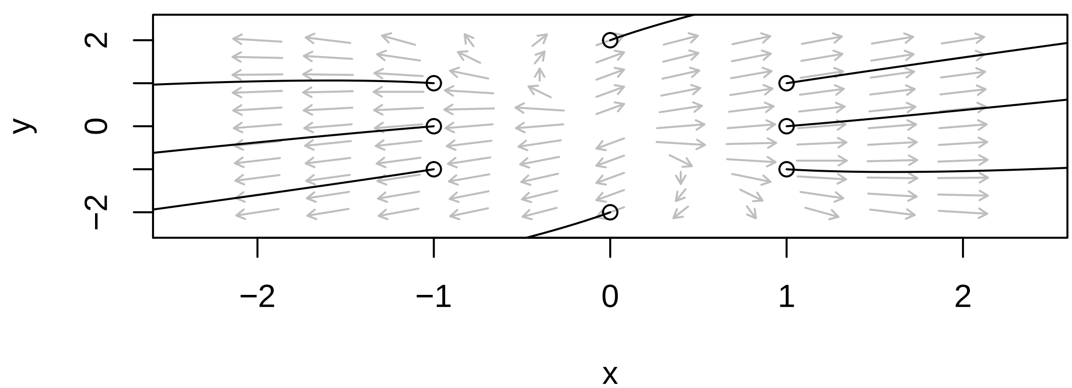
(a)

$$\dot{x} = 6x + 2y$$
$$\dot{y} = 2x + 3y$$

```
## [1] "Phase portrait type:  source"
```

Therefore we expect to have two distinct real eigenvalues:

```
## [1] "Eigenvalues:  7" "Eigenvalues:  2"
```



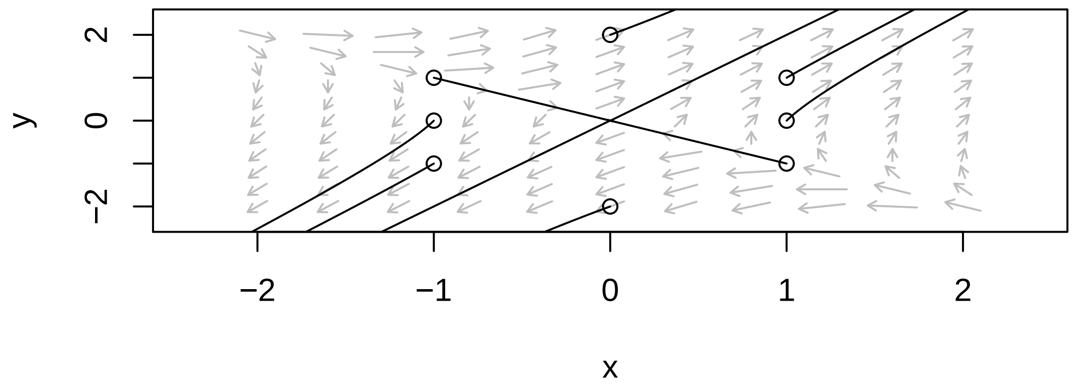
(b)

$$\dot{x} = x + 2y$$
$$\dot{y} = 4x + 3y$$

```
## [1] "Phase portrait type:  saddle"
```

Therefore we expect two distinct real eigenvalues:

```
## [1] "Eigenvalues:  5"  "Eigenvalues:  -1"
```



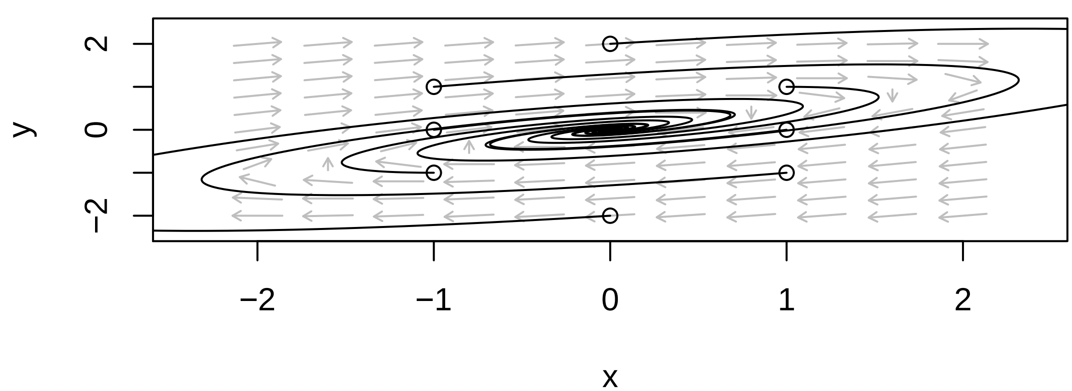
(c)

$$\dot{x} = -2x + 4y$$
$$\dot{y} = -x + y$$

```
## [1] "Phase portrait type:  spiral sink"
```

Therefore we expect a complex conjugate pair of eigenvalues:

```
## [1] "Eigenvalues:  -0.5+1.3228756555323i"
## [2] "Eigenvalues:  -0.5-1.3228756555323i"
```



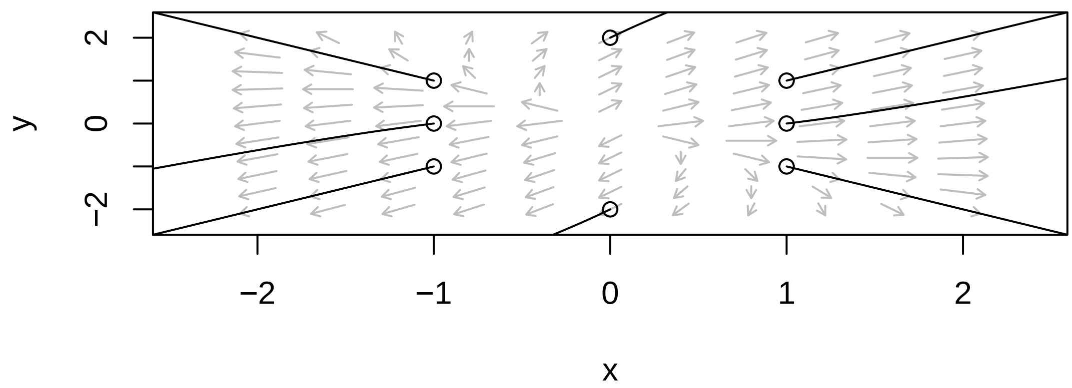
(d)

$$\dot{x} = 2x + y$$
$$\dot{y} = x + 2y$$

```
## [1] "Phase portrait type:  source"
```

Therefore we expect two distinct real eigenvalues:

```
## [1] "Eigenvalues:  3" "Eigenvalues:  1"
```



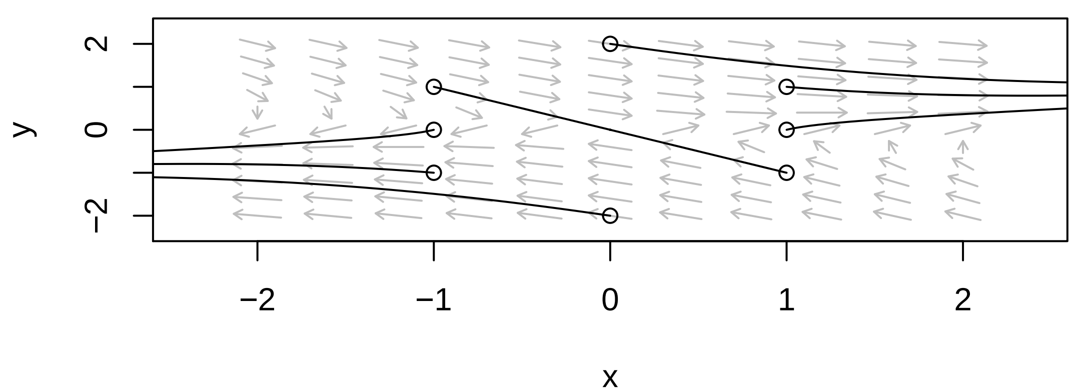
(e)

$$\dot{x} = x + 5y$$
$$\dot{y} = x - 3y$$

```
## [1] "Phase portrait type:  saddle"
```

Therefore we expect two distinct real eigenvalues:

```
## [1] "Eigenvalues:  -4" "Eigenvalues:  2"
```



(f)

$$\dot{x} = -1x + ay$$
$$\dot{y} = 0x + ay$$
$$\text{for } a \neq 0$$

This function changes behavior, dependent on the value of a, when it crosses $det(f) = 0, tr(f) = 0, det(f) = \frac{tr(f)^2}{4}$ lines on the trace-determinant plane. So I will find these critical values alegbraically.

$$tr(f) = -1 + a = 0 \quad \rightarrow \quad \boxed{a = 1}$$
$$a = 1 \text{ is not critical value since } det(d) < 0$$
$$det(f) = -a = 0 \quad \rightarrow \quad \boxed{a = 0}$$
$$delta = det(f) - \frac{tr(f)^2}{4} = -a - \frac{(a-1)^2}{4}$$
$$= a^2 + 2a + 1 \quad \rightarrow \quad \boxed{a = -1}$$

Classification and visualization for $a = -1.5, -1.0, -0.5, 0, 0.5$:

```
## [1] "sink"
```

```
## [1] "degenerate sink"
```

```
## [1] "sink"
```

```
## [1] "stable line"
```

```
## [1] "saddle"
```

Therefore we expect the following eigenvalue combinations (in order):
-Distinct real
-One real
-Distinct real
-Distinct real
-Distinct real

```
## [1] "Eigenvalues:  -1.5" "Eigenvalues:  -1"
```

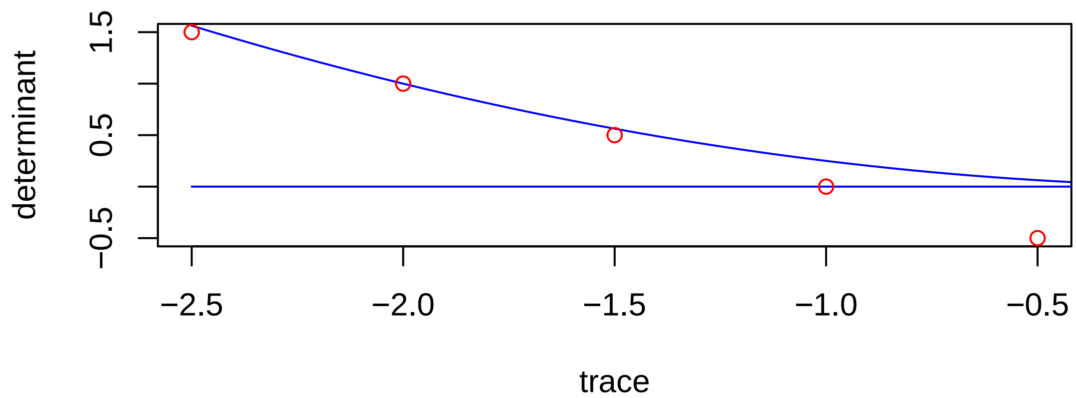
```
## [1] "Eigenvalues:  -1" "Eigenvalues:  -1"
```

```
## [1] "Eigenvalues:  -1"   "Eigenvalues:  -0.5"
```

```
## [1] "Eigenvalues:  0"  "Eigenvalues:  -1"
```

```
## [1] "Eigenvalues:  -1"  "Eigenvalues:  0.5"
```

```
fs <- list(f_n1.5,f_n1.0,f_n0.5,f_p0.0,f_p0.5)
deters <- lapply(fs,det)
traces <- lapply(fs,function(A){sum(diag(A))})
vals <- seq(from = -2.5,to=2.5,length.out=101)
plot(vals,vals^2/4,col="blue",
     xlim=c(-2.5,-.5),ylim=c(-0.5,1.5),
     type="l",xlab="trace",ylab="determinant")
points(vals,rep(0,101),col="blue",type="l")
points(traces,deters,col="red")
```



Therefore, the system exhibits the following behavior:
Sink for $(-\infty < a < -1) \cup (-1, 0)$
Degenerate sink for $a = -1$
Stable line for $a = 0$
Saddle for $(0 < a < \infty)$

Problem #2

```
plotSystem <- function(func,center,params=c())
{
  x <- center[1]; y <- center[2];
  ff  <- flowField(func,
                   xlim = c(x-2, x+2), ylim = c(y-2,y+2),
                   parameters = params,
                   points = 11,add = FALSE)
  state <- matrix(c(x+1,y+1,x+1,y-1,x-1,y+1,x-1,y-1,
                    x+0,y+2,x+0,y-2,x-1,y+0,x+1,y+0,
                    x+2,y+2,x+2,y-2,x-2,y+2,x-2,y-2)
                  ,12, 2, byrow = TRUE)
  trajs <- trajectory(func,y0 = state, tlim = c(0, 10),
                      parameters = params,add=TRUE)
}
```

(a)

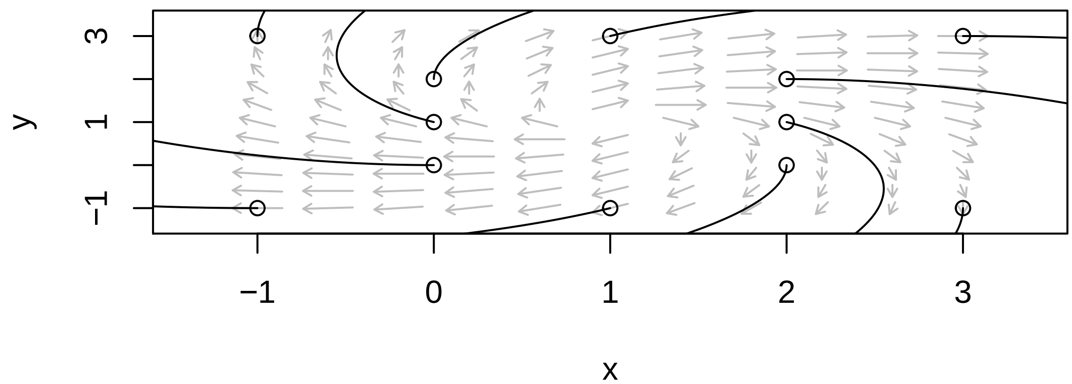
$$\dot{x} = x + y - 2, \quad \dot{y} = y - x$$
$$x = y + 2$$
$$x = y$$

Algebraically there are no steady states.

```
  a2 <- function(t,state,parameters){
    with(as.list(c(state,parameters)),{
    dx <- state[1] + state[2]-2
```

```
    dy <- -state[1] + state[2]
    list(c(dx,dy))
    })}
center <- c(1,1)
plotSystem(a2,center)
```

```
## Note: col has been reset as required
```



(b)

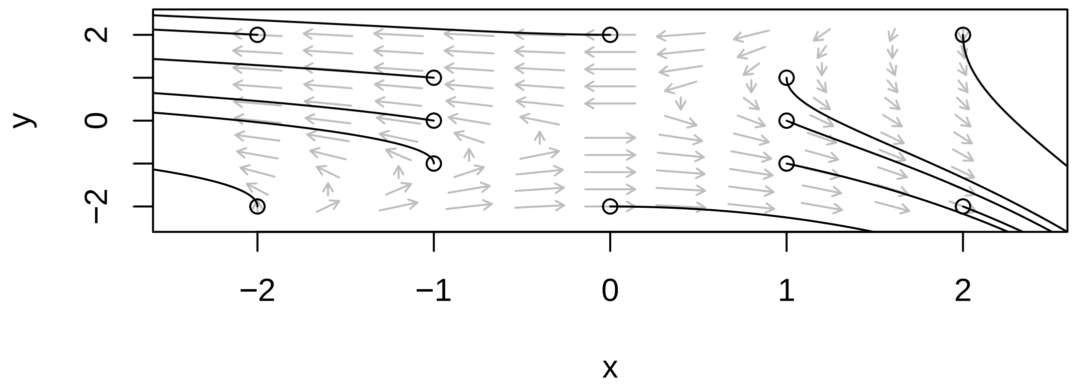
$$\dot{x} = x - y, \quad \dot{y} = 1 - e^x$$

$$x = y$$
$$x = \ln(1) = 0$$

Therefore, algebraic steady state at $(x, y) = (0, 0)$.

```
  b2 <- function(t,state,parameters){
    with(as.list(c(state,parameters)),{
    dx <- state[1] - state[2]
    dy <- 1-exp(state[1])
    list(c(dx,dy))
    })}
center <- c(0,0)
plotSystem(b2,center)
```

```
## Note: col has been reset as required
```



(c)

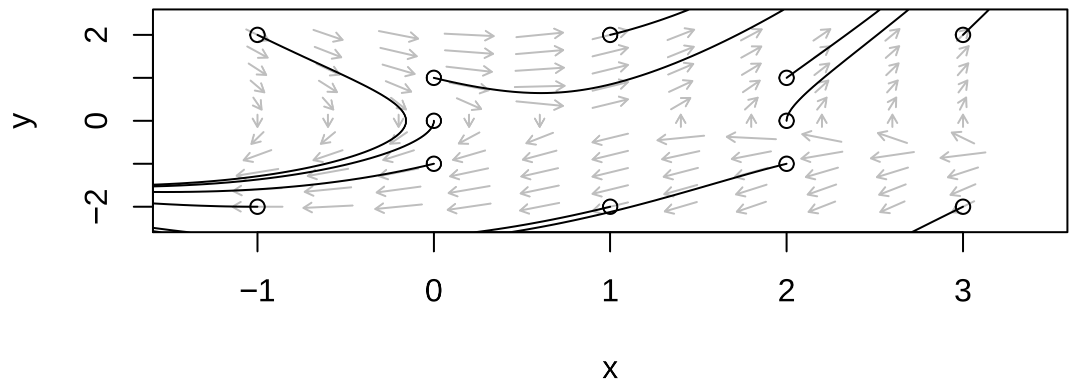
$$\dot{x} = y, \quad \dot{y} = x(1+y) - 1$$

$$y = 0$$
$$y = \frac{1}{x} - 1$$

Therefore, algebraic steady state at $(x, y) = (1, 0)$.

```
c2 <- function(t,state,parameters){
    with(as.list(c(state,parameters)),{
    dx <- state[2]
    dy <- state[1]*(1+(state[2]))-1
    list(c(dx,dy))
    })}
center <- c(1,0)
plotSystem(c2,center)
```

```
## Note: col has been reset as required
```



(d)

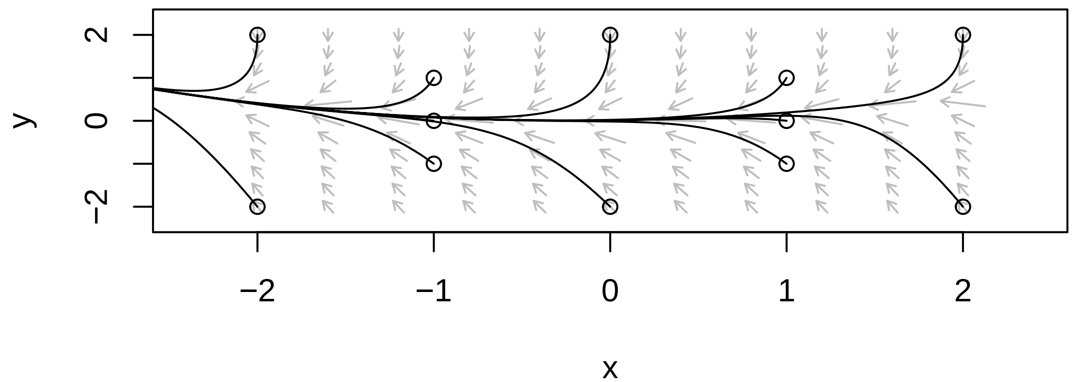
$$\dot{x} = y - 2, \quad \dot{y} = x^2 - 8y$$

$$y = 2$$
$$y = \frac{x^2}{8}$$

Therefore, algebraic steady state at $(x, y) = (4, 2)$.

```
  d2 <- function(t,state,parameters){
    with(as.list(c(state,parameters)),{
    dx <- state[2]-2
    dy <- state[1]^2-8*state[2]
    list(c(dx,dy))
    })}
center <- c(0,0)
plotSystem(d2,center)
```

```
## Note: col has been reset as required
```



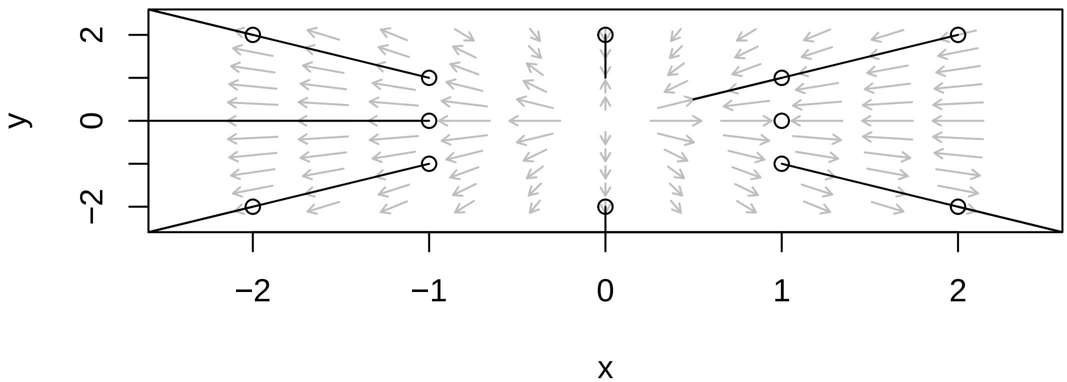
(e)

$$\dot{x} = (\lambda - ax - by)x, \quad \dot{y} = (\mu - cx - dy)y$$

$$x = 0, \quad y = 0$$
$$0 = \lambda - ax - by$$
$$0 = \mu - cx - dy$$

```
  e2 <- function(t,state,params){
    with(as.list(c(state,params)),{
    dx <- (l-a*state[1]-b*state[2])*state[1]
    dy <- (u-c*state[1]-d*state[2])*state[2]
    list(c(dx,dy))
    })}
center <- c(0,0); parameters <- c(l=1,u=1,a=1,b=1,c=1,d=1)
plotSystem(e2,center, params = parameters)
```

```
## Note: col has been reset as required
```



2
0
−2
y
−2
−1
0
1
2
x