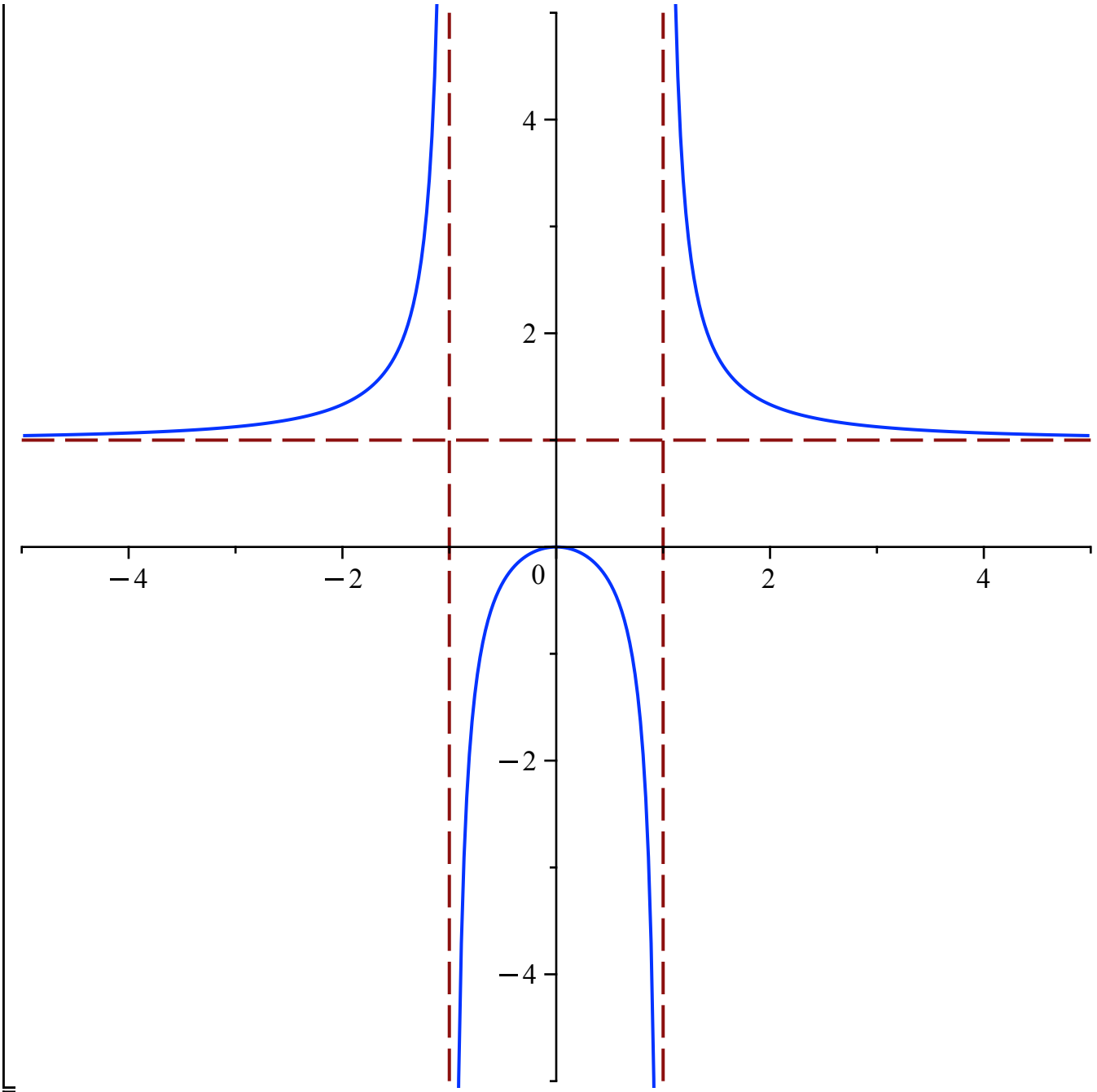
```
> restart
> with(plottools):
> with(plots):
> with(LinearAlgebra):
> with(DEtools):
> currentdir("/Users/matej/Documents/vscht/maple")
```

"/Users/matej" **(1)**

1)

```
> f := x -> x**2/((x+1)*(x-1)) :
> g_f := plot(f, view=[-5..5, -5..5], color=blue):
> as_1 := plot([1, t,t=-5..5], line_style=Dash):
> as_2 := plot([-1, t,t=-5..5], line_style=Dash):
> as_3 := plot([t, 1, t=-5..5], line_style=Dash):
> display(g_f, as_1, as_2, as_3)
```




2)

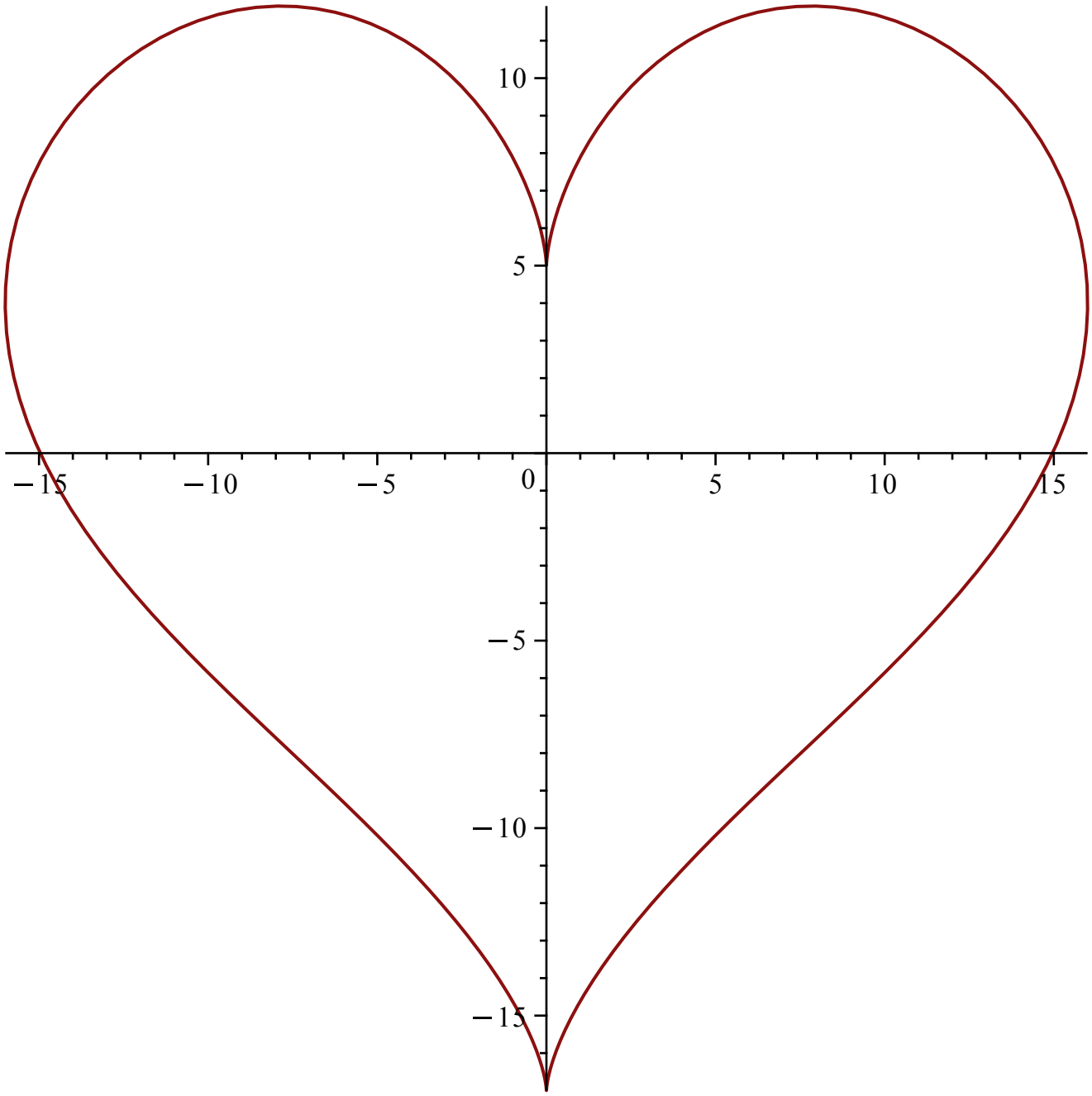
```
> A_ := Matrix(4, 4, [-4, 3, 1, 7, 1, -1, 1, -6, 8, -2, 3, 0, 4, 0,
  5, 1]):
> b := Vector([2, -7, 17, 11]):
> x_:= LinearSolve(A_,b)
```

$$x\_ := \begin{bmatrix} 1 \\ -3 \\ 1 \\ 2 \end{bmatrix} \quad \textbf{(2)}$$

3)

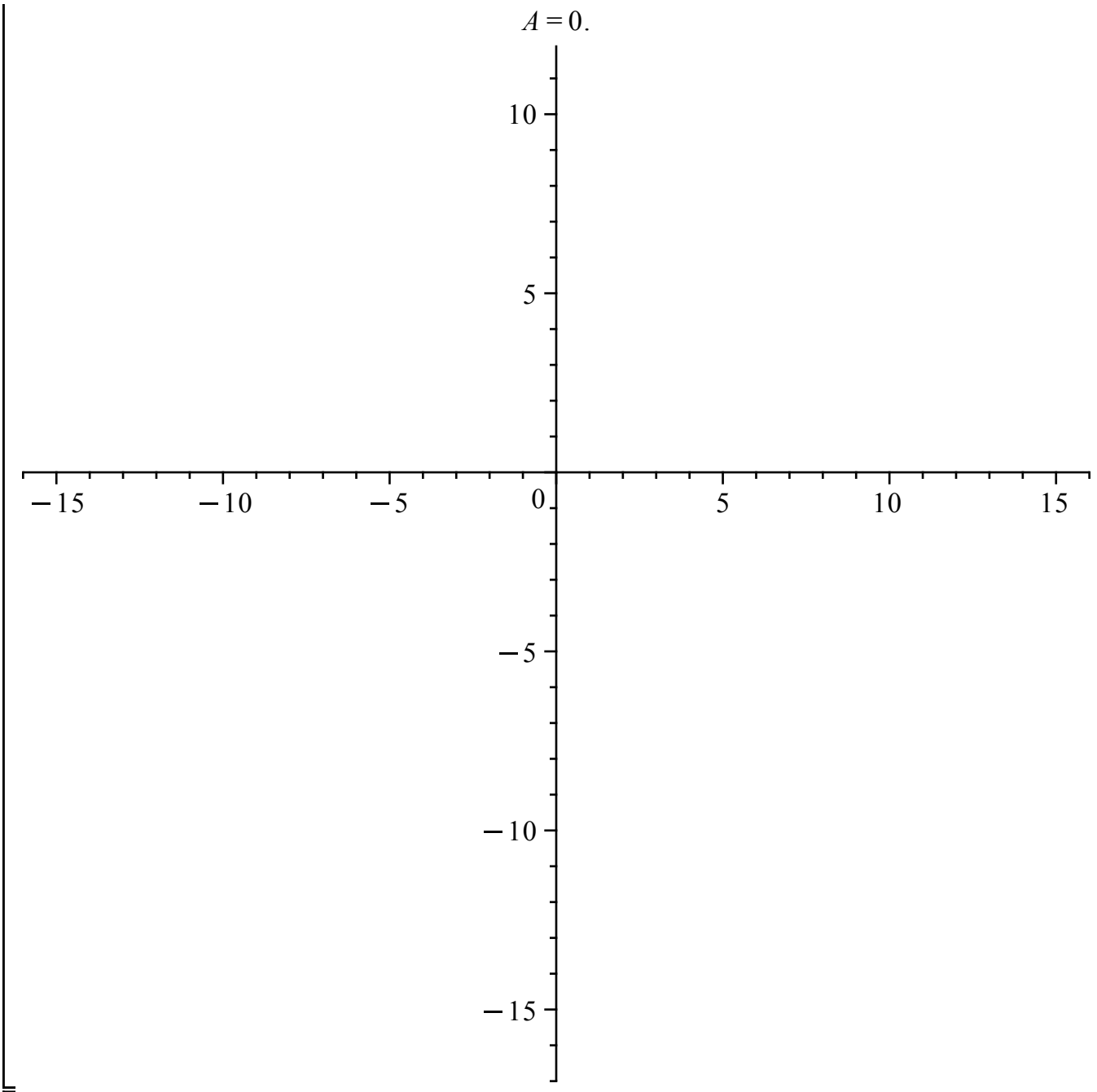
a)

```
> k := plot([16*sin(t)**3, 13*cos(t)-5*cos(2*t)-2*cos(3*t)-cos(4*t),
  t=0..2*Pi])
```




b)

```
> animate(plot, [[16*sin(t)**3, 13*cos(t)-5*cos(2*t)-2*cos(3*t)-cos
  (4*t), t=0..A]], A=0..2*Pi)
```




c)

```
> l := Int(
      sqrt(
          (diff(16*sin(t)^3, t))^2 +
          (diff(13*cos(t) - 5*cos(2*t) - 2*cos(3*t) - cos(4*t), t))^2
      ),
      t = 0 .. 2*Pi
  ):
> delka := evalf(l)
```

$$delka := 102.1675516 \quad \textbf{(3)}$$

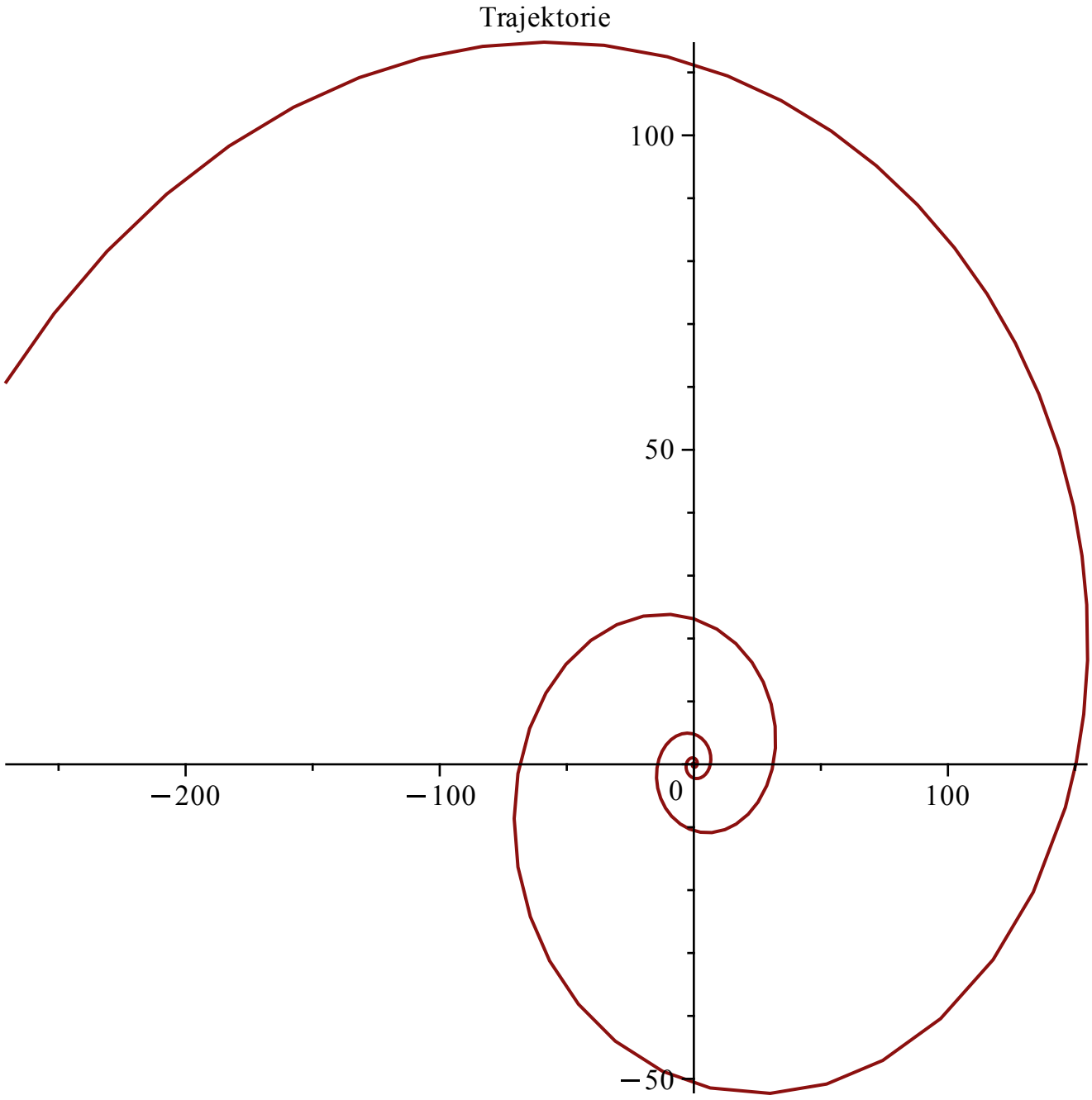
4)

```
> odes := diff(x(t), t) = x(t) - 8*y(t), diff(y(t), t) = 2*x(t) + y
```

```
  (t):
> ics := x(0) = 0, y(0)=1:
> reseni := dsolve([odes, ics], [x(t), y(t)])
```

$$reseni := \{x(t) = -2\,e^{t}\sin(4\,t), y(t) = e^{t}\cos(4\,t)\} \quad \textbf{(4)}$$

```
> trajektorie:=plot([rhs(reseni[1]), rhs(reseni[2]), t=-5..5], title=
  "Trajektorie")
```

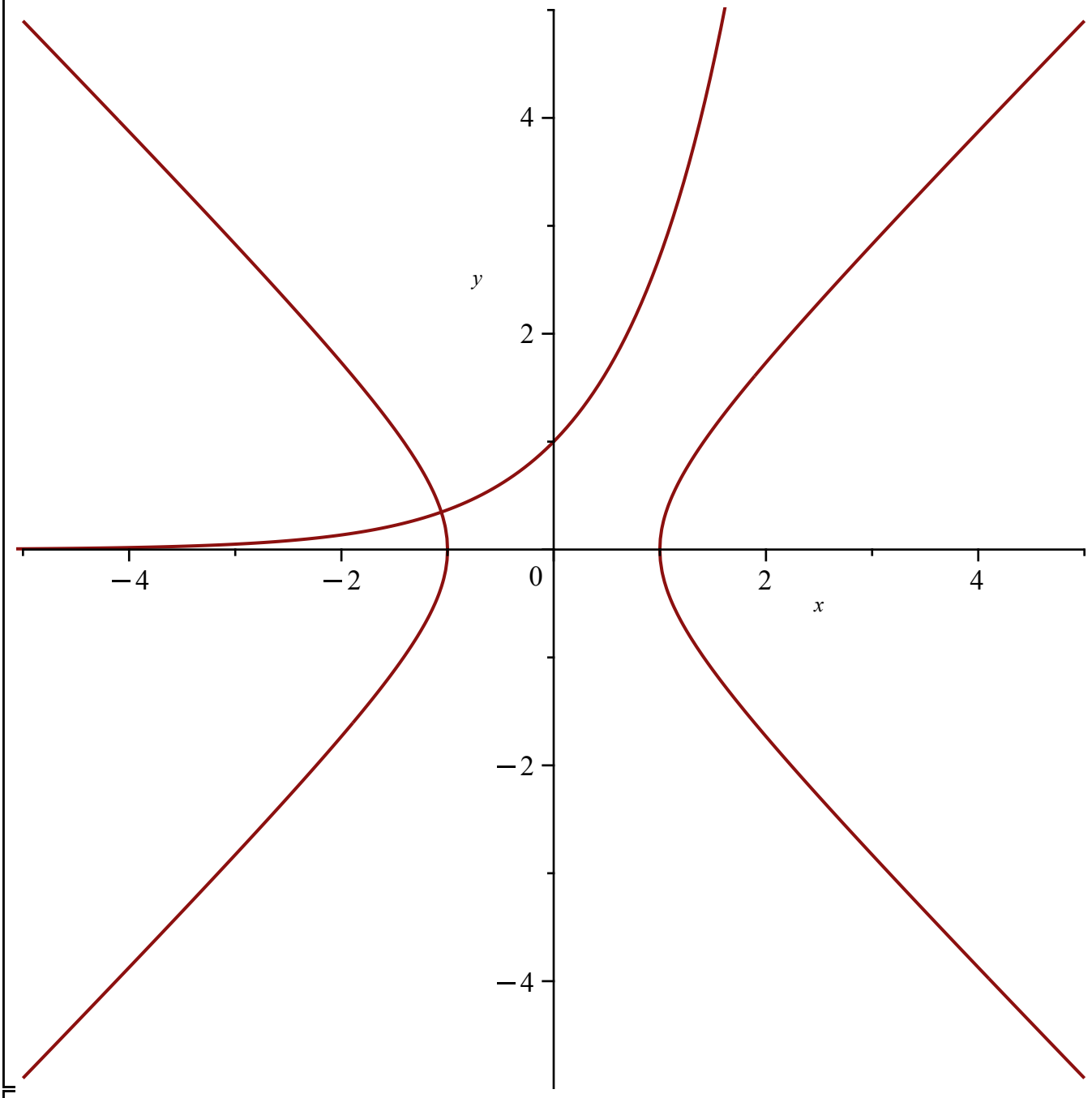



5)

a)

```
> r1 := implicitplot(x**2-y**2-1=0, x=-5..5, y=-5..5):
> r2 := plot(exp(x)):
```

```
> display(r1, r2, view=[-5..5,  -5..5])
```




b)

=> 1 reseni (realne)

```
> restart
> reseni := evalf(allvalues(solve([x^2 - y^2 - 1 = 0, y - exp(x) =
  0], [x, y]))):
> realne_reseni := op(op(reseni[1]))
```

$$realne\_reseni := x = -1.058487282,\ y = 0.3469802969 \qquad \textbf{(5)}$$

6)

a)

```
> ode := diff(y(x),x) = x*exp(-y(x))
```

$$ode := \frac{d}{dx} y(x) = x\, e^{-y(x)} \quad (6)$$

```
> ic := y(0) = 0:
> solution := dsolve([ode, ic], [y(x)])
```

$$solution := y(x) = -\ln(2) + \ln(x^2 + 2) \quad (7)$$

```
> y(2) = evalf(subs(x=2, rhs(solution)))
```

$$y(2) = 1.098612288 \quad (8)$$

b)

```
EulerMethod := proc(f, x0, y0, h, xmax)
    local x, y, n, yn, i;
    #n := convert((xmax - x0) / h, integer); #pocet iteraci
    x := x0; y := y0; # poc podminka
    for i from x0 to xmax by h do
        yn := y + h * f(x, y);
        x := x + h;
        y := yn;
    od;
    return y;
end proc:
```

```
> f := (x,y) -> x*exp(-y)
```

$$f := (x, y) \mapsto x \cdot e^{-y} \quad (9)$$

```
> EulerMethod(f, 0, 0, 0.1, 2); # pro krok 0.1
> EulerMethod(f, 0, 0, 0.01, 2); # pro krok 0.1
```

$$1.154104852$$

$$1.104113833 \quad (10)$$

Eulerova metoda hazi reseni, ktere je vetsi -> jde tam z prava