MATH 151 Lab 6

Put team members' names and section number here.

Kevin Lei Jonathan Madding Pan Zhou John Schumacher

Section number 576

In [1]:
```
from sympy import *
from sympy.plotting import (plot,plot_parametric)
```

Question 1

1a

In [2]:
```
x, r = symbols("x, r")
y = E ** (r * x)

print("r = ", end = "")
print(solve((2 * diff(y, x, 2) + diff(y, x, 1) - y), r))
```

```
r = [-1, 1/2]
```

1b

In [3]:
```
print("r = ", end = "")
print(solve((diff(y, x, 2) + 6 * diff(y, x, 1) + 10 * y), r))
```

```
r = [-3 - I, -3 + I]
```

1c

In [4]:
```
y = E ** (-3 * x) * (cos(x) + sin(x))
print((diff(y, x, 2) + 6 * diff(y, x, 1) + 10 * y))
print("The cos(x) + sin(x) part of the equation allows it to be a solution of the diff
```

```
6*(-sin(x) + cos(x))*exp(-3*x) - 8*(sin(x) + cos(x))*exp(-3*x) + 2*(7*sin(x) + cos
(x))*exp(-3*x)
The cos(x) + sin(x) part of the equation allows it to be a solution of the differenti
al equation without having an imaginary exponent.
```

Question 2

2a

In [5]:
```
t = symbols("t")
x = E ** (2 * sin(t))
y = E ** cos(t)

print(f"<{x.subs(t, pi / 6) + diff(x, t, 1).subs(t, pi / 6) * t}, {y.subs(t, pi / 6) +
print(f"<{N(x.subs(t, pi / 6)) + diff(x, t, 1).subs(t, pi / 6) * t}, {N(y.subs(t, pi /
```

```
<sqrt(3)*E*t + E, -t*exp(sqrt(3)/2)/2 + exp(sqrt(3)/2)>
<sqrt(3)*E*t + 2.71828182845905, -t*exp(sqrt(3)/2)/2 + 2.37744267523616>
```

2b

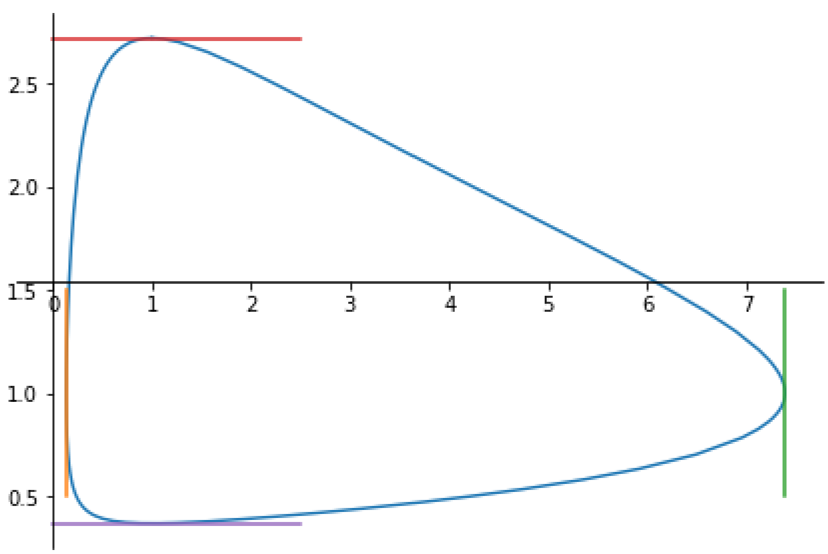
In [6]:
```
print("Horizontal:")
for i in solve(diff(y, t, 1)):
    print(f"({x.subs(t, i)}, {y.subs(t, i)})")

print("Vertical:")
for i in solve(diff(x, t, 1)):
    print(f"({x.subs(t, i)}, {y.subs(t, i)})")
```

```
Horizontal:
(1, E)
(1, exp(-1))
Vertical:
(exp(2), 1)
(exp(-2), 1)
```

2c

In [7]:
```
import math
t = symbols("t")
x = E ** (2 * sin(t))
y = E ** cos(t)
plot0 = plot_parametric((x, y), (t, 0, 2 * pi), show = false)
p_vert1=plot_parametric((math.e**-2,t,(t,.5,1.5)),(math.e**2,t,(t,.5,1.5)),show=False)
p_horiz1=plot_parametric((t,math.e,(t,0,2.5)),(t,math.e**-1,(t,0,2.5)),show=False)
plot0.extend(p_vert1)
plot0.extend(p_horiz1)
plot0.show()
```

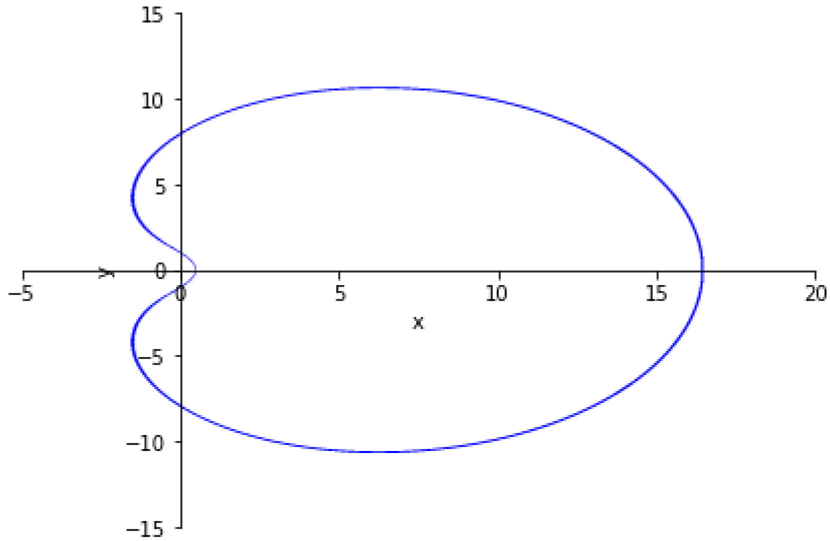


Question 3

3a

In [8]:
```
x, y = symbols("x, y")
limacon = (-1 * ((x ** 2 + y ** 2) / 4) + 2 * x - 2) ** 2 - 5 * (x ** 2 + y ** 2)
```

```
plot_implicit(limacon, (x, -5, 20), (y, -15, 15))
```



Out[8]: <sympy.plotting.plot.Plot at 0x12b6e93d4c0>

3b

In [9]:
```
print(idiff(limacon, y, x))
```

```
(-x**3 + 12*x**2 - x*y**2 + 4*y**2 + 32)/(y*(x**2 - 8*x + y**2 - 32))
```

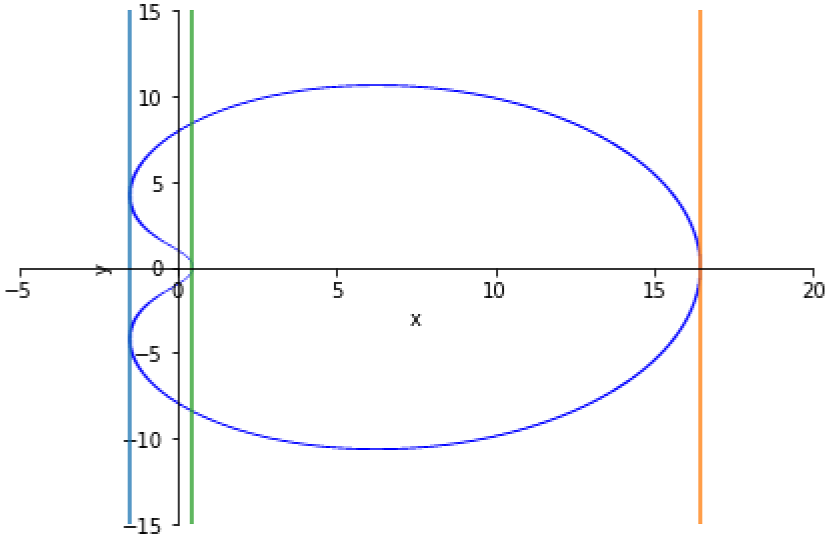
3c

In [10]:
```
from sympy import *
from sympy.plotting import (plot, plot_parametric)
x = symbols('x')
y = symbols('y')
limacon = (-1 * ((x**2 + y ** 2) / 4) + 2 * x - 2) ** 2 - 5 * (x ** 2 + y ** 2)
dydx = idiff(limacon, y, x)
den = denom(dydx)
vtan = solve([den, limacon],[x, y])
vtan2 = solve(limacon.subs(y, 0))
print(f'Vertical tangents at {vtan[0]}, {vtan[1]}, ({vtan2[0]}, 0), ({vtan2[1]}, 0)')
```

```
Vertical tangents at (-3/2, -sqrt(71)/2), (-3/2, sqrt(71)/2), (4 + 2*sqrt(5) + 2*sqrt
(7 + 4*sqrt(5)), 0), (-2*sqrt(5) + 4 - 2*sqrt(7 - 4*sqrt(5)), 0)
```

3d

In [11]:
```
#3d
pcurve=plot_implicit(limacon,(x,-5,20),(y,-15,15),show=False)
t=symbols('t')
p_vert=plot_parametric((-3/2,t,(t,-15,15)),(4 + 2*sqrt(5) + 2*sqrt(7 + 4*sqrt(5)),t,(t
p_vert2 = plot_parametric(.47,t,(t,-15,15), show = False)
pcurve.extend(p_vert)
pcurve.extend(p_vert2)
pcurve.show()
```



Question 4

4a

In [12]:
```
x = symbols("x")
y = (x ** (1 / 5) * (x ** 3 + 1) ** (1 / 2)) / (2 - 7 * x) ** 4

expanded = expand_log(ln(y), force = True)
print(diff(expanded, x, 1) * y)
```

```
x**0.2*(x**3 + 1)**0.5*(1.5*x**2/(x**3 + 1) + 28/(2 - 7*x) + 0.2/x)/(2 - 7*x)**4
```

4b

In [13]:
```
print(diff(y, x, 1))
```

```
0.2*(x**3 + 1)**0.5/(x**0.8*(2 - 7*x)**4) + 28*x**0.2*(x**3 + 1)**0.5/(2 - 7*x)**5 +
1.5*x**2.2/((2 - 7*x)**4*(x**3 + 1)**0.5)
```

4c

In [28]:
```
print((diff(expanded, x, 1) * y).expand())
print((diff(y, x, 1)).expand())

print((diff(expanded, x, 1) * y).expand() == (diff(y, x, 1)).expand())
```

```
28*x**0.2*(x**3 + 1)**0.5/(-16807*x**5 + 24010*x**4 - 13720*x**3 + 3920*x**2 - 560*x
+ 32) + 1.5*x**2.2/(2401*x**4*(x**3 + 1)**0.5 - 2744*x**3*(x**3 + 1)**0.5 + 1176*x**2
*(x**3 + 1)**0.5 - 224*x*(x**3 + 1)**0.5 + 16*(x**3 + 1)**0.5) + 0.2*(x**3 + 1)**0.5/
(16*x**0.8 - 224*x**1.8 + 1176*x**2.8 - 2744*x**3.8 + 2401*x**4.8)
28*x**0.2*(x**3 + 1)**0.5/(-16807*x**5 + 24010*x**4 - 13720*x**3 + 3920*x**2 - 560*x
+ 32) + 1.5*x**2.2/(2401*x**4*(x**3 + 1)**0.5 - 2744*x**3*(x**3 + 1)**0.5 + 1176*x**2
*(x**3 + 1)**0.5 - 224*x*(x**3 + 1)**0.5 + 16*(x**3 + 1)**0.5) + 0.2*(x**3 + 1)**0.5/
(16*x**0.8 - 224*x**1.8 + 1176*x**2.8 - 2744*x**3.8 + 2401*x**4.8)
True
```