MATH 151 Lab 8

Hudson hurtig Jax Lanier Carson Kjar Ashton hull

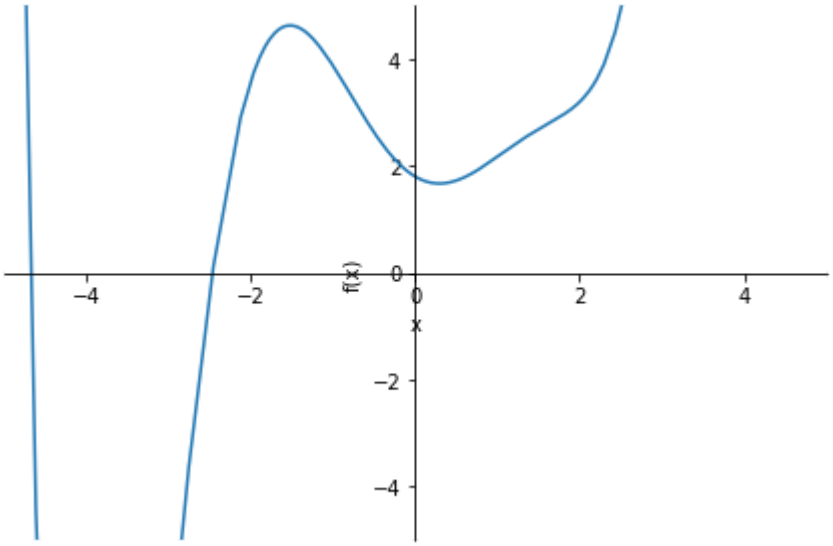
In [2]:
```
from sympy import *
from sympy import Symbol, N
from sympy.plotting import (plot,plot_parametric)
```

Question 1

1a

In [3]:
```
x = symbols('x', real=True)


def isIncreasing(func, p):

    funcPrime = diff(func,x,1)

    func2Prime = diff(func,x,2)

    if func2Prime.subs(x,p) > 0: # since the equation is lienar this function technica
        return True # a derrivative is basically a limit which considers nums around i
    return False


f1 = (1/40) * (x**6 + 2*x**5 - 16*x**4 + 64*x**2 -36*x +72)

fprime = diff(f1,x,1)



sols = solve(fprime)


for i in sols:
    if isIncreasing(f1, i):
        print(f"f is increasing at x = {i}")
    else:
        print(f"f is decreasing at x = {i}")

plot(f1,xlim=[-5,5], ylim=[-5,5])
```

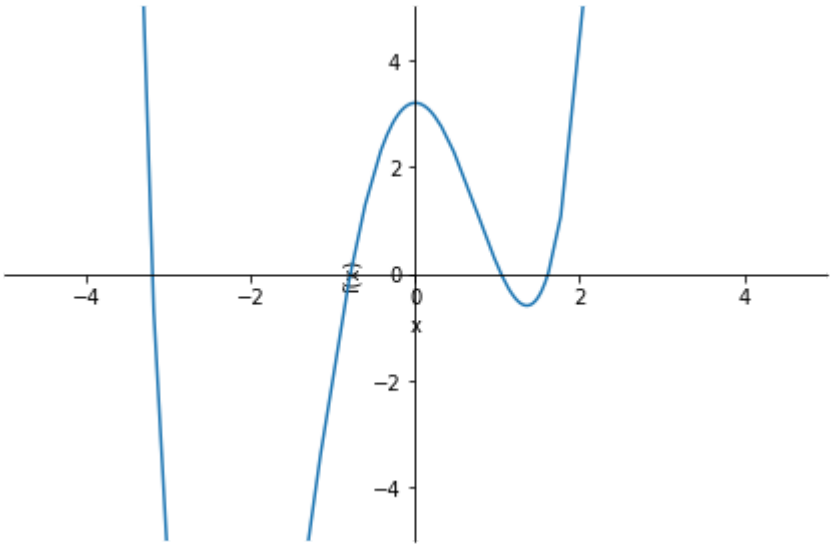
```
f is increasing at x = -3.98318299325190
f is decreasing at x = -1.52373005204977
f is increasing at x = 0.293170162363288
```



Out[3]: <sympy.plotting.plot.Plot at 0x24fb48c6640>

1b

In [4]:
```
plot(diff(f1,x,2),xlim=[-5,5], ylim=[-5,5])


#print([nsolve(diff(f1,x,2),i) for i in range(-10,10)])
inflections = [*set([nsolve(diff(f1,x,2),i) for i in range(-10,10)])]
#print(inflections)


for i in inflections:
    if i > 0:  # a derrivative is basically a limit which considers nums around it
        print(f"f is concave up at x = {i}")
    else:
        print(f"f is concave down at x = {i}")
```




```
f is concave up at x = 1.04315333651963
f is concave down at x = -0.790426944270573
f is concave up at x = 1.61601988903000
f is concave down at x = -3.20207961461240
```

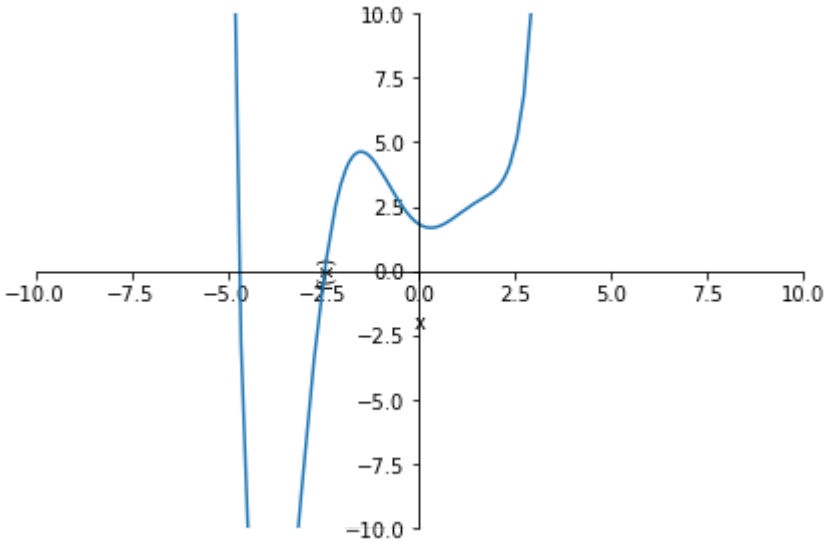
1c

In [5]:
```
print(f"a total of {len(inflections)+len(sols)} inflection points and extrrema exist")
```

```
a total of 7 inflection points and extrrema exist
```

1d

In [6]:
```
plot(f1,xlim=[-10,10], ylim=[-10,10])
print("verified")
```

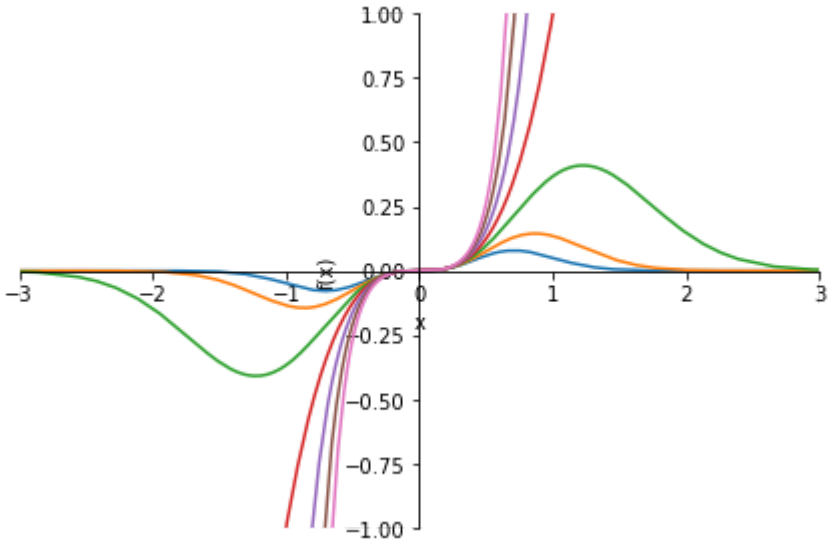



```
verified
```

Question 2

2a

In [7]:
```
x = symbols('x',real=True)

b = symbols('b',real=True)

g = x**3 * exp(b*x**2)

eqs = [g.subs(b,i) for i in range(-3,4)]

nums = [-3,-2,-1,0,1,2,3]



plot(eqs[0],eqs[1],eqs[2],eqs[3],eqs[4],eqs[5],eqs[6],xlim=[-3,3], ylim=[-1,1])
```



Out[7]: <sympy.plotting.plot.Plot at 0x24fb73e2d30>

2b

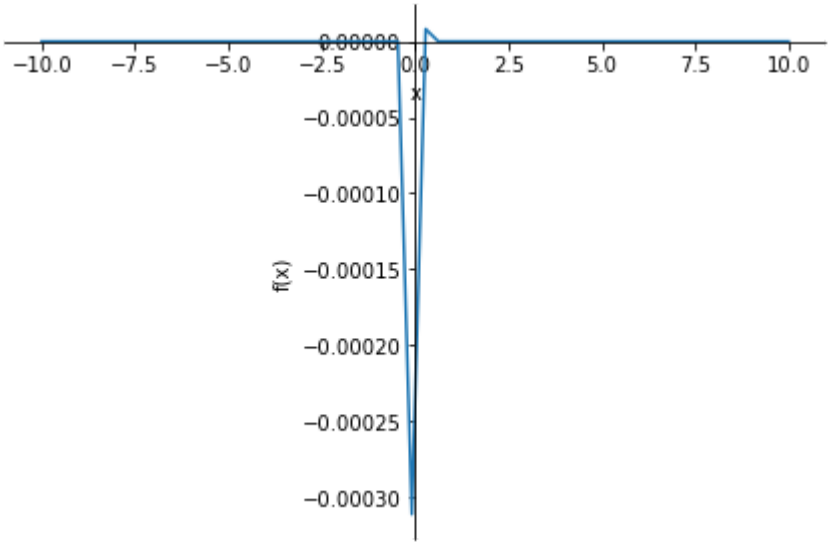
In [8]:
```
for i in range(len(eqs)):
    print(f"the critical values of g where b = {nums[i]}, are {solve(diff(eqs[i],x))}"
```

```
the critical values of g where b = -3, are [0, -sqrt(2)/2, sqrt(2)/2]
the critical values of g where b = -2, are [0, -sqrt(3)/2, sqrt(3)/2]
the critical values of g where b = -1, are [0, -sqrt(6)/2, sqrt(6)/2]
the critical values of g where b = 0, are [0]
the critical values of g where b = 1, are [0]
the critical values of g where b = 2, are [0]
the critical values of g where b = 3, are [0]
```

2c

In [9]:
```
print("as b approaches -infinity we see that we gain more and more prominate extrema t
plot(g.subs(b,-100))
```

```
as b approaches -infinity we see that we gain more and more prominate extrema that ge
t closer and closer to where x = 0
```



Out[9]: <sympy.plotting.plot.Plot at 0x24fb744df10>

2d

In [10]:
```
print("all inlfection points are real at x = 0 regaurdless of the value of b")
```

```
all inlfection points are real at x = 0 regaurdless of the value of b
```

2e

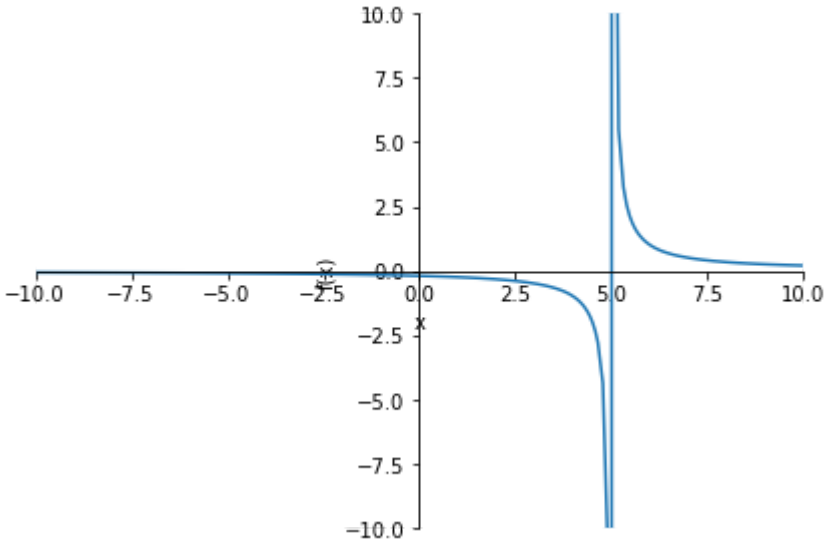
In [40]:
```
print("b needs to equal this fraction of x: ", solve(diff(g,x) - 1))
```

```
b needs to equal this fraction of x:  [{b: (LambertW(-exp(3/2)/(2*x**2)) - 3/2)/x**
2}, {b: (LambertW(exp(3/2)/(2*x**2)) - 3/2)/x**2}]
```

Question 3

3a

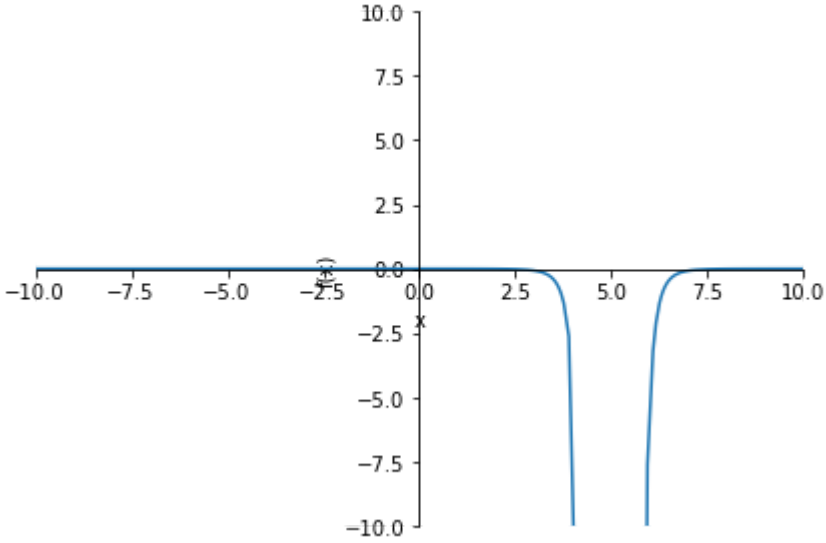
In [55]:
```
def findAvgSlope(eqs,x1,x2):
    plot(diff(eqs),xlim=[-10,10], ylim=[-10,10])
    return (eqs.subs(x,x2) - eqs.subs(x,x1))/(x2-x1)

def confirmSlope(eqs, slope):

    return solve(diff(eqs) - slope)

print("MVT states ln(5-x) over the interval [1,4] avg slope is :",N(findAvgSlope(ln(5-
```




```
MVT states ln(5-x) over the interval [1,4] avg slope is : -0.462098120373297 we can c
onfirm that this slope falls within the range given because - 1/2 falls within our ra
nge referencing the graph
```

3b

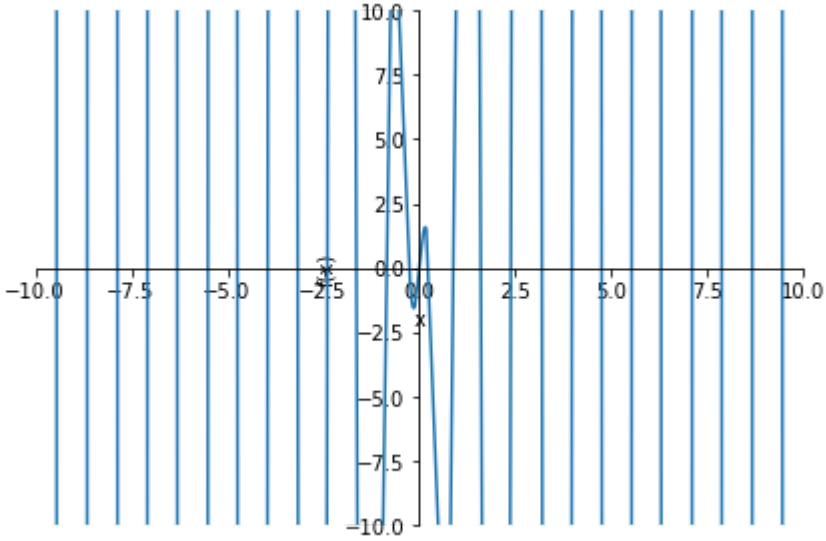
In [56]:
```
print("MVT states (x-5)^-5 over the interval [0,8] avg slope is :",N(findAvgSlope((x-5
```



MVT states (x-5)^-5 over the interval [0,8] avg slope is : 0.000554403292181070 we ca n confirm that this slope falls within the range given because a slope of nearly zero falls within our range referencing the graph

3c

In [57]:
```
print("MVT states 8*x**2*cos(4*x) over the interval [0,8] avg slope is :",N(findAvgSlo
```



MVT states 8*x**2*cos(4*x) over the interval [0,8] avg slope is : -25.1196814105034 w e can confirm that this slope falls within the range given because a slope of -25 fal ls within our range referencing the graph

In [ ]: