MATH 151 Lab 8

Jaeheon Kim, Nico Bui, Alex Dao, Vishwa Kumaravel

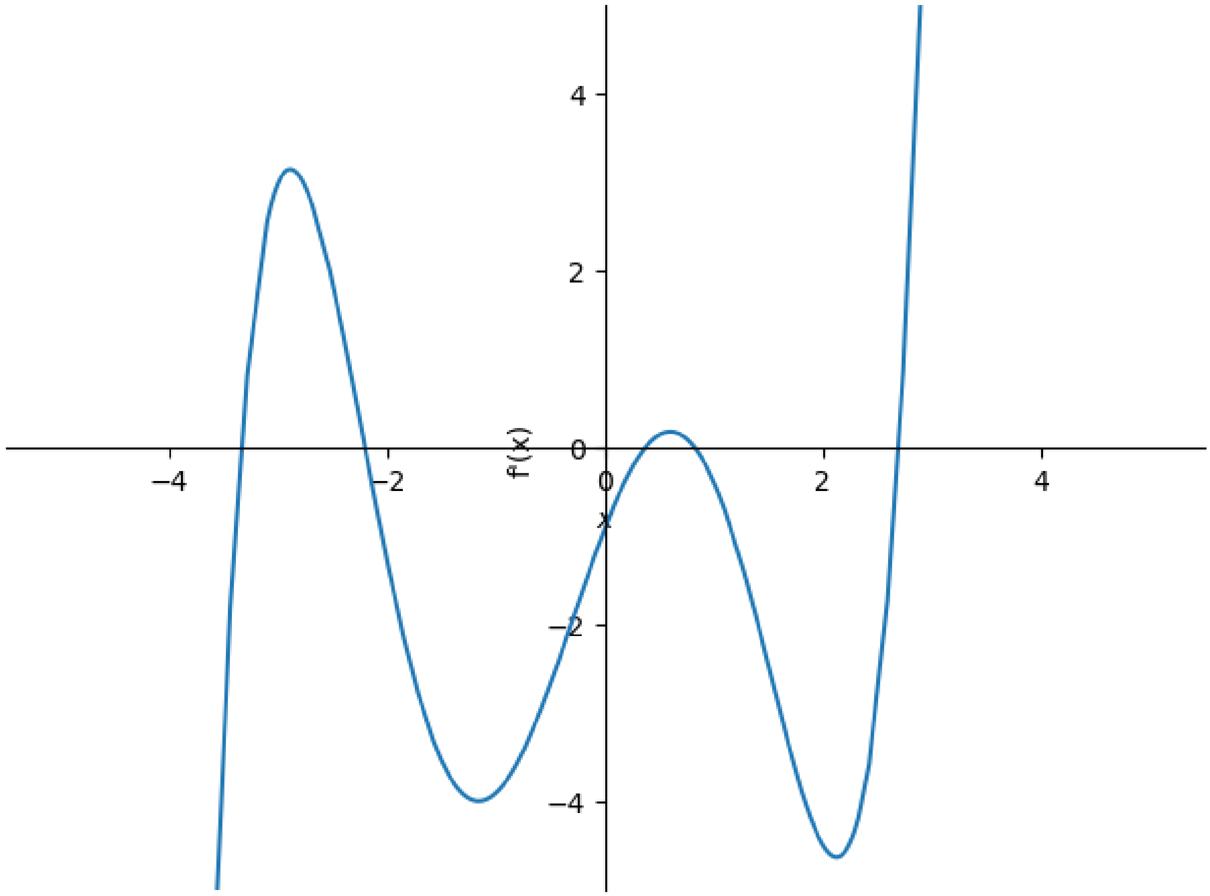
In [4]:
```
from sympy import *
from sympy.plotting import (plot,plot_parametric)
```

Question 1

1a

In [83]:
```
from sympy import Symbol, diff, solveset
from sympy.plotting import plot
import numpy as np
import pandas as pd

# (a)

# solve to get f'(x)
x = Symbol('x')
f_x = 1/40*(x**6+2*(x**5)-16*(x**4)-20*(x**3)+64*(x**2)-36*x+72)
first_deriv = diff(f_x, x)
print("f'(x):", first_deriv, end='\n')

# solve f'(x)=0 to get critical values
critical_values=list(solveset(first_deriv, x))
print('Critical Values:', critical_values, end='\n')

# take a value in each range to check whether it is increasing or decreasing in that r
critical_values_rows = []
for i in range(len(critical_values)):
    if i == 0:
        x_val = critical_values[i] - 1
        lb, ub = -np.inf, critical_values[i]
    elif i == len(critical_values) - 1:
        x_val = critical_values[i] + 1
        lb, ub = critical_values[i], np.inf
    else:
        x_val = (critical_values[i-1] + critical_values[i])/2
        lb, ub = critical_values[i-1], critical_values[i]

    func_val = first_deriv.subs(x, x_val)
    critical_values_rows.append([lb, ub, 'Increasing' if func_val>0 else 'Decreasing']

critical_values_df = pd.DataFrame(critical_values_rows,
                                  columns=['Lower Bound', 'Upper Bound', 'Increasing/D
print(critical_values_df, end='\n')

# plot f'(x)
fplot=plot(first_deriv,(x,-5,5), ylim=(-5, 5), ylabel="f'(x)")
```

```
f'(x): 0.15*x**5 + 0.25*x**4 - 1.6*x**3 - 1.5*x**2 + 3.2*x - 0.9
Critical Values: [-3.34365086397455, -2.20571930723638, 0.367785714582751, 0.82115699
8770767, 2.69376079119075]
         Lower Bound         Upper Bound Increasing/Decreasing
0               -inf   -3.34365086397455            Decreasing
1  -3.34365086397455   -2.20571930723638            Increasing
2  -2.20571930723638   0.367785714582751            Decreasing
3  0.367785714582751   0.821156998770767            Increasing
4   2.69376079119075                 inf            Increasing
```



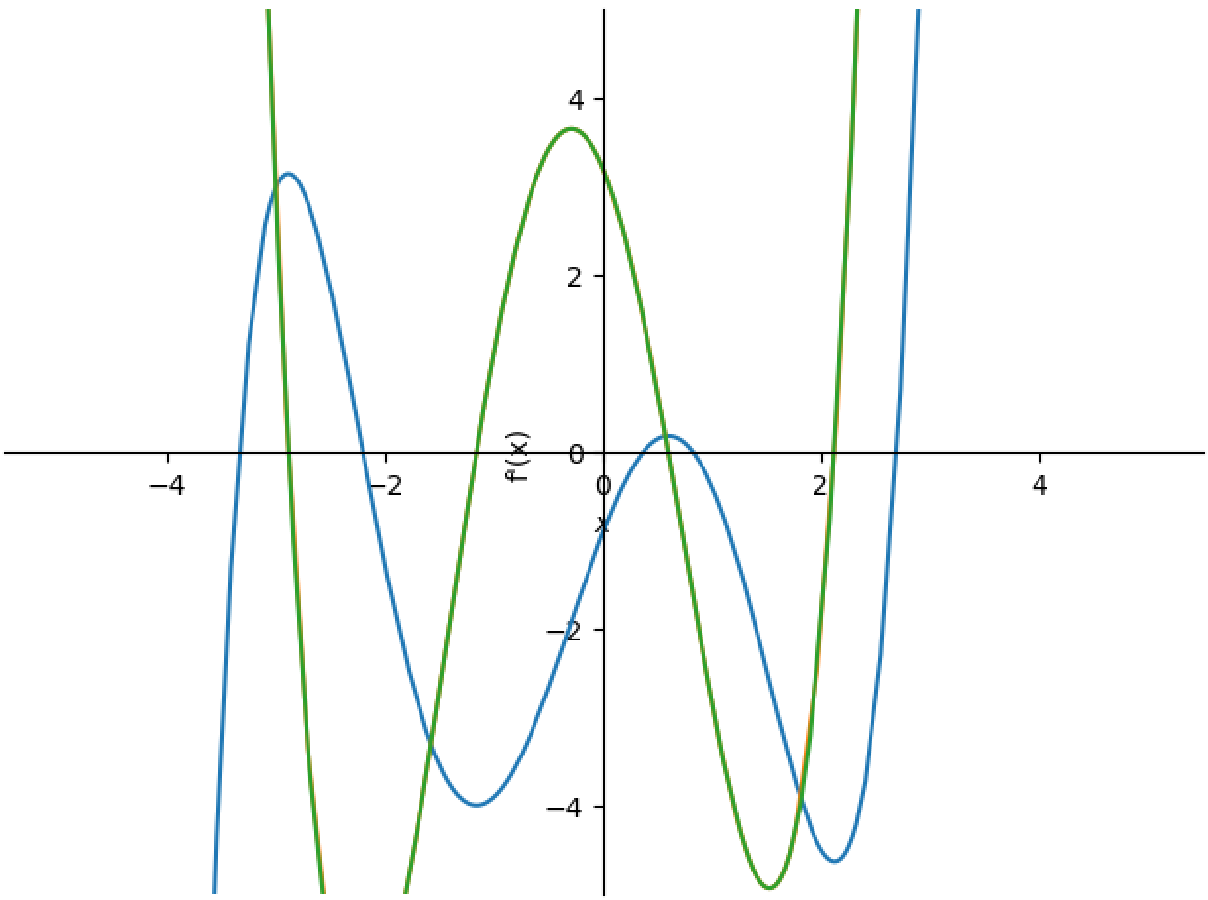
1b

In [3]:
```
matplotlib notebook
```

In [85]:
```
# solve to get f''(x)
second_deriv = diff(f_x, x, 2)
print("f''(x):", second_deriv, end='\n')

# solve f''(x)=0 to get inflection values
inflection_values=list(solveset(second_deriv, x))
print('Inflection Values:', inflection_values, end='\n')

inflection_values_rows = []
for i in range(len(inflection_values)):
    if i == 0:
        x_val = inflection_values[i] - 1
        lb, ub = -np.inf, inflection_values[i]
    elif i == len(inflection_values) - 1:
        x_val = inflection_values[i] + 1
        lb, ub = inflection_values[i], np.inf
```

```
    else:
        x_val = (inflection_values[i-1] + inflection_values[i])/2
        lb, ub = inflection_values[i-1], inflection_values[i]

    func_val = second_deriv.subs(x, x_val)
    inflection_values_rows.append([lb, ub, 'Up' if func_val>0 else 'Down'])

inflection_values_df = pd.DataFrame(inflection_values_rows,
                                    columns=['Lower Bound', 'Upper Bound', 'Concave Up
print(inflection_values_df, end='\n')
fpplot = plot(second_deriv,(x,-5,5), ylim=(-5, 5), ylabel="f''(x)",show = False)
fpplot.extend(fpplot)
fpplot.show()
```

```
f''(x): 0.75*x**4 + 1.0*x**3 - 4.8*x**2 - 3.0*x + 3.2
Inflection Values: [-2.89174218338126, -1.16242859299527, 0.597894461879547, 2.122942
98116364]
          Lower Bound        Upper Bound Concave Up/Down
0                -inf  -2.89174218338126              Up
1   -2.89174218338126  -1.16242859299527            Down
2   -1.16242859299527  0.597894461879547              Up
3    2.12294298116364                inf              Up
```



1c

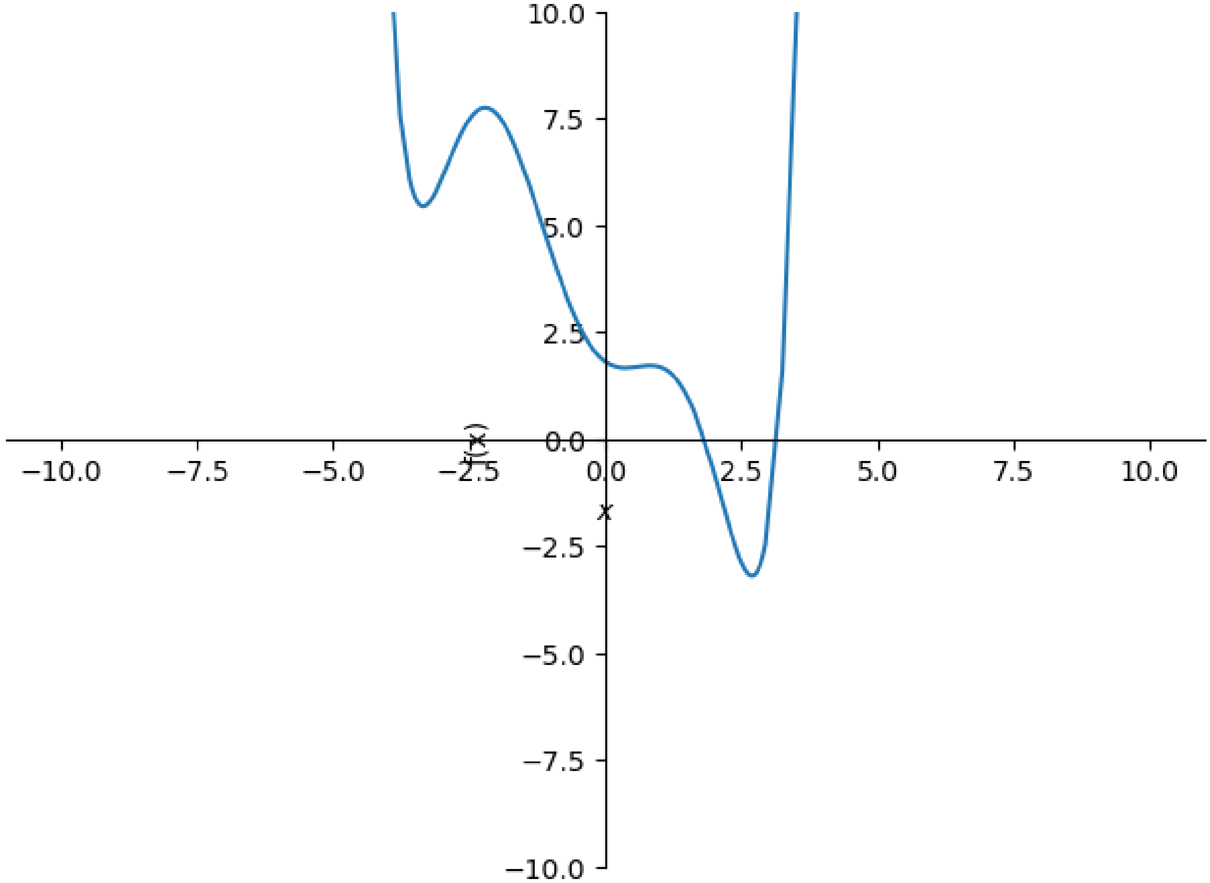
In [87]:
```
print("For part (c),using the graph and information from part(a) and part(b), Number c
```

```
For part (c),using the graph and information from part(a) and part(b), Number of loca
l extrema: 5 Number of inflection points: 4
```

1d

In [6]:
```
matplotlib notebook
```

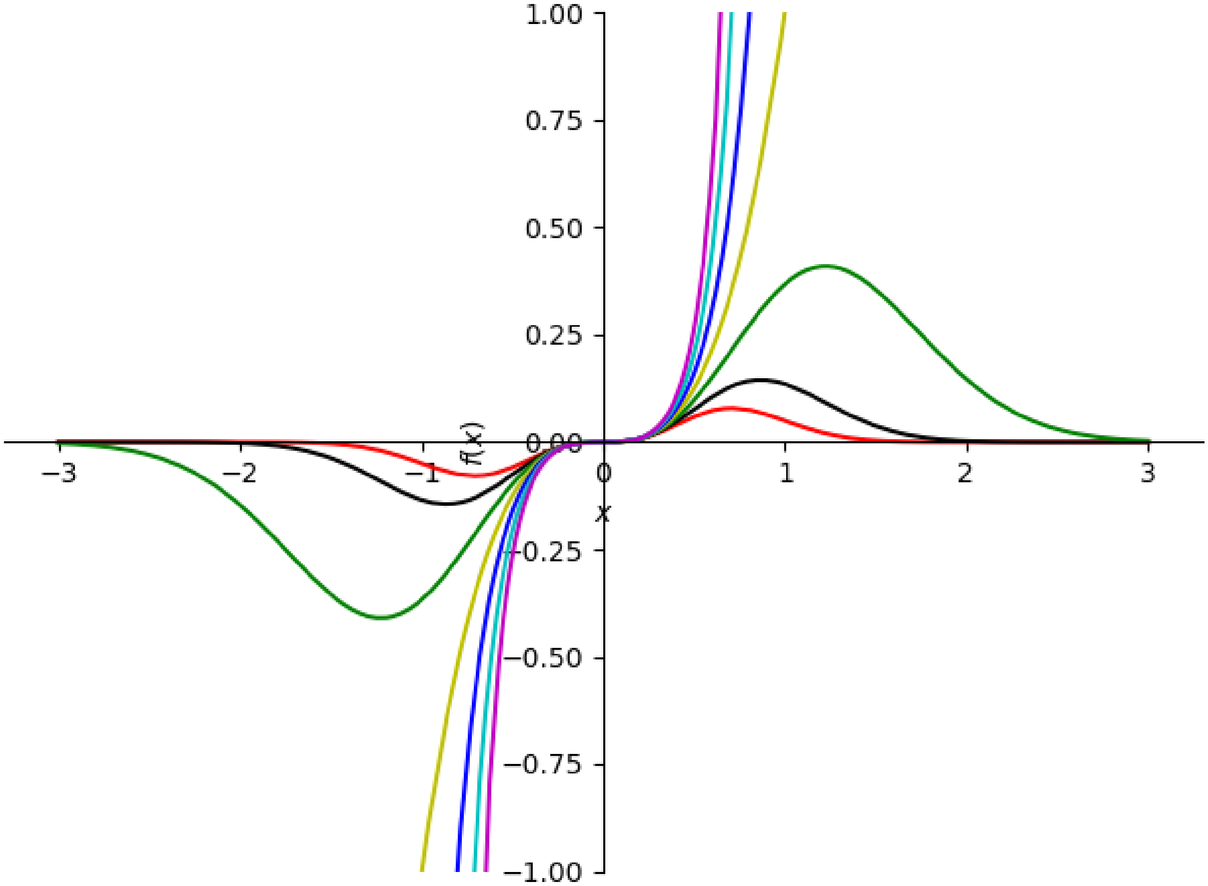
In [86]:
```
fplottt=plot(f_x,(x,-10,10), ylim=(-10, 10), ylabel="f(x)")
```



Question 2

2a

In [49]:
```
x,b = symbols('x b')
g = x**3*E**(b*x**2)
domain = [-3,-2,-1,0,1,2,3]
bColors = {-2: 'k', -1: 'g', 0: 'y',1:'b',2:'c',3:'m'}
Plt = plot(g.subs(b,-3),(x,-3,3),ylim = [-1,1],line_color='r',show=False)
for i in domain[1:]:
    Pltt = plot(g.subs(b,i),(x,-3,3),ylim = [-1,1],line_color=bColors[i],show=False)
    Plt.extend(Pltt)
Plt.show()
```



2b

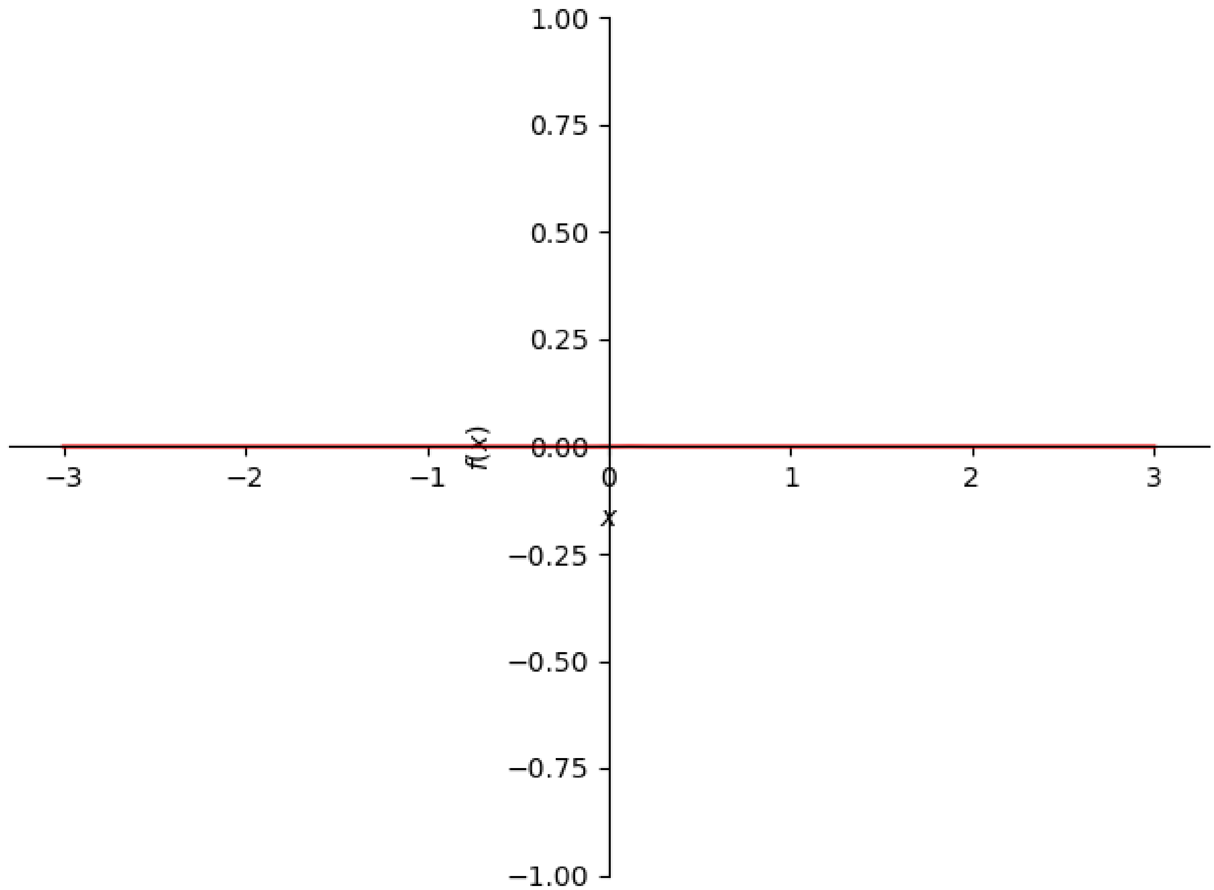
In [52]:
```
gp = diff(g,x)

cp = solve(gp,x)
print(f'A critical values in terms of b is{cp}')
print('If b=0, -1/b is undefined, 1 critical point, If b>0, -1/b is negative, no criti
print('So critical values are all real when b = -3,-2, and -1')
```

```
A critical values in terms of b is[0, -sqrt(6)*sqrt(-1/b)/2, sqrt(6)*sqrt(-1/b)/2]
If b=0, -1/b is undefined, 1 critical point, If b>0, -1/b is negative, no critical po
ints, If b<0, -1/b is positive, all 3 critical points are real
So critical values are all real when b = -3,-2, and -1
```

2c

In [ ]:
```
matplotlib notebook
```

In [44]:
```
Plt = plot(g.subs(b,-100),(x,-3,3),ylim = [-1,1],line_color='r',show=False)
Plt.show()
cp = diff(g.subs(b,-100),x)
cpp = solve(cp,x)
print(f'As critical values at b =-100 are {cpp} and the graph seems to be becoming fla
```



As critical values at b =-100 are [0, -sqrt(6)/20, sqrt(6)/20] and the graph seems to be becoming flatter, critical values approaches 0 as b -> -oo

2d

In [53]:
```
ip = diff(g,x,2)
ipp = solve(ip,x)
print(ip)
print(f'inflection points: {ipp}')
print('If b=0, there is a line, no inflection points, If b>0, -1/b is negative, so x=0
print('So All inflection points are real when b = -3,-2,-1')
```

```
2*x*(b*x**2*(2*b*x**2 + 1) + 6*b*x**2 + 3)*exp(b*x**2)
inflection points: [0, -sqrt(2)*sqrt(-1/b)/2, sqrt(2)*sqrt(-1/b)/2, -sqrt(3)*sqrt(-1/
b), sqrt(3)*sqrt(-1/b)]
If b=0, there is a line, no inflection points, If b>0, -1/b is negative, so x=0 is th
e only inflection point, If b<0, -1/b is positive, so all 5 inflection points are rea
l
So All inflection points are real when b = -3,-2,-1
```

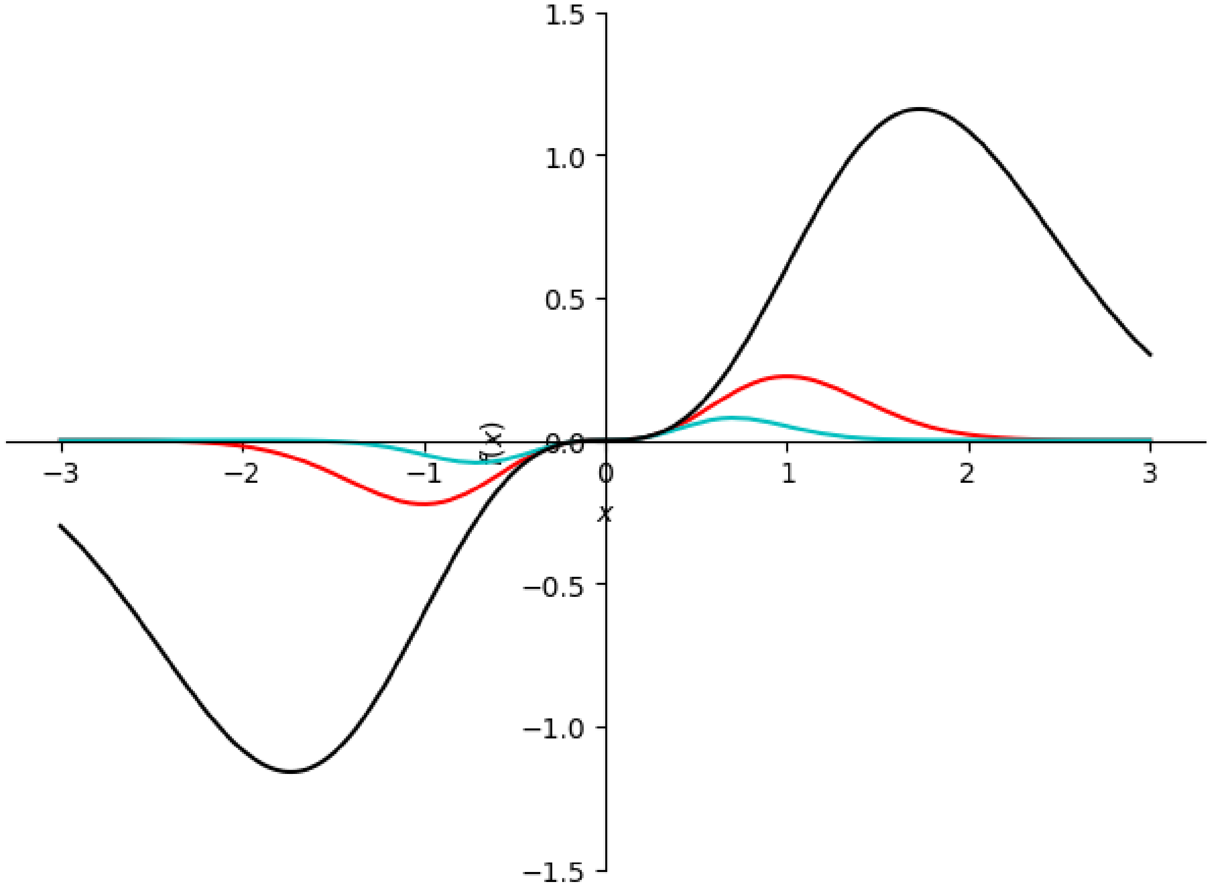
2e

In [ ]:
```
matplotlib notebook
```

In [69]:
```
gp = diff(g,x)
gpp = diff(g,x,2)
cp = solve(gp.subs(x,-1),b)
cpp = solve(gp.subs(x,1),b)
ip = solve(gpp.subs(x,-1),b)
ipp = solve(gpp.subs(x,1),b)
```

```
print(f'Since {cp}(when x is -1) and {cpp}(when x is 1) are the same, critical values
print(f'Since {ip}(when x is -1) and {ipp}(when x is 1) are the same, inflection point
plt1 = plot(g.subs(b,cp[0]),(x,-3,3),ylim = [-1.5,1.5],line_color='r',show=False)
plt2 = plot(g.subs(b,ip[0]),(x,-3,3),ylim = [-1.5,1.5],line_color='c',show=False)
plt3 = plot(g.subs(b,ip[1]),(x,-3,3),ylim = [-1.5,1.5],line_color='k',show=False)
plt1.extend(plt2)
plt1.extend(plt3)
plt1.show()
```

```
Since [-3/2](when x is -1) and [-3/2](when x is 1) are the same, critical values incl
ude +-1 when b is [-3/2]
Since [-3, -1/2](when x is -1) and [-3, -1/2](when x is 1) are the same, inflection p
oints include +-1 when b is -3 or -1/2
```



Question 3

3a

In [72]:
```
from sympy import *
from sympy.plotting import (plot,plot_parametric)
x = symbols('x')
fx = (1 + x) / (1 + x**2)
fx_prime = diff(fx, x)
fx_2_prime = diff(fx_prime, x)
inf_pts_x = solve(fx_2_prime, x)
inf_pts_y = []
for pt in inf_pts_x:
    inf_pts_y.append(fx.subs(x, pt))
slope = (inf_pts_y[1] - inf_pts_y[0]) / (inf_pts_x[1] - inf_pts_x[0])
x1 = inf_pts_x[0]
```

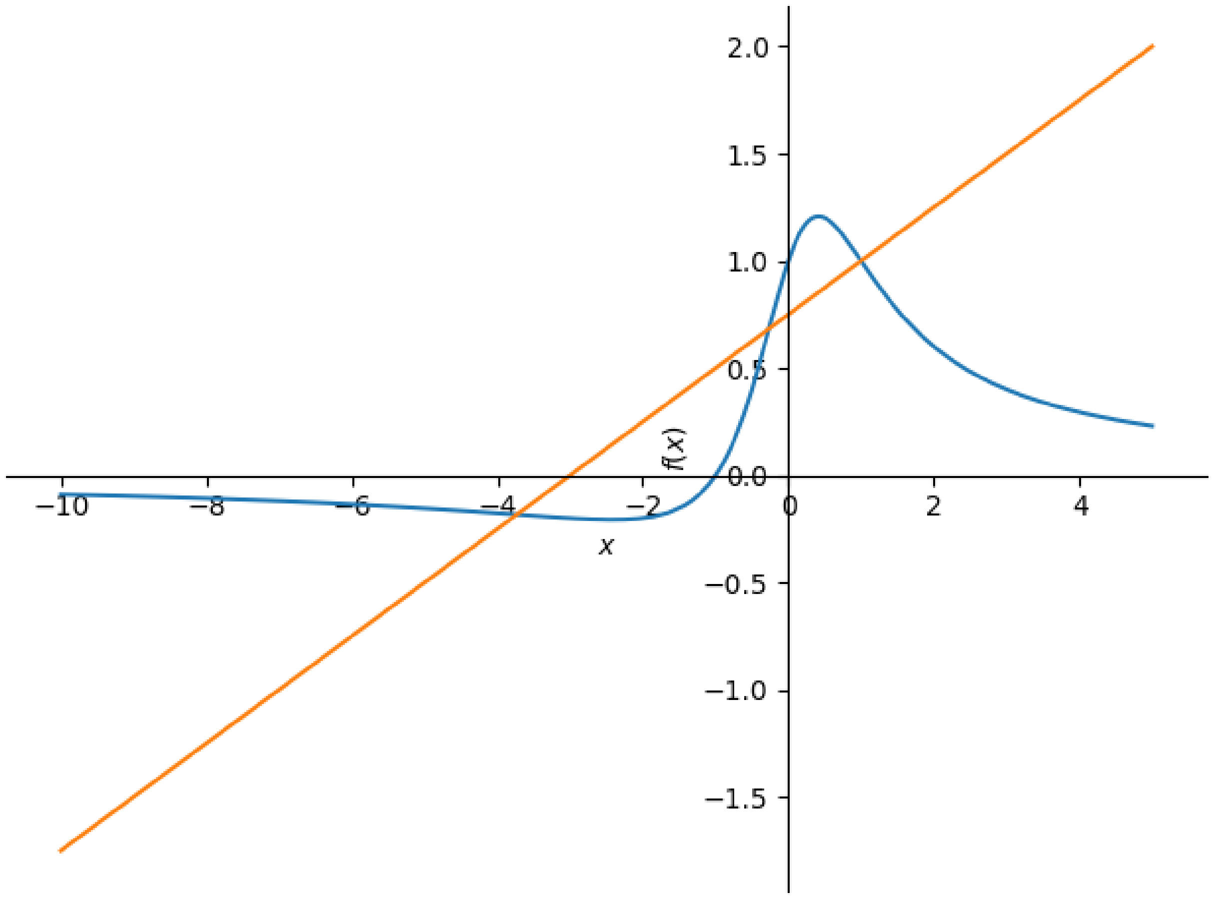
```
y1 = inf_pts_y[0]
fin_eq = slope * (x - x1) + y1
print(f" y = {fin_eq}")
```

```
 y = (-1 + (-sqrt(3) - 1)/(1 + (-2 - sqrt(3))**2))*(x - 1)/(-3 - sqrt(3)) + 1
```

3b

In [ ]:
```
matplotlib notebook
```

In [73]:
```
plot(fx, fin_eq,(x,-10,5))
```



Out[73]:
```
<sympy.plotting.plot.Plot at 0x2343051e910>
```

In [ ]: