Name	UIN
Huy Lai	132000359
Alexander Nuccitelli	000000000
Cole Jahnke	530009075

MATH 151-557

09 November 2021

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

1a find and simplify the derivative of f-g

In [11]:
```
x = symbols('x', positive=True)
f = 2 * asin(x)
g = acos(1 - 2 * x ** 2)
h = (f - g).diff(x)
print("The derivative is", h)
print("Simplified, it becomes,", simplify(h), "when x is non-negative")
```

```
The derivative is -4*x/sqrt(1 - (1 - 2*x**2)**2) + 2/sqrt(1 - x**2)
Simplified, it becomes, 0 when x is non-negative
```

1bc Implication of f-g and specific answer

In [13]:
```
print("When x is non-negative, f(x) - g(x) is 0, implying that f(x) and g(x) are equiva
print("f(0) - g(0) =", f.subs(x, 0) - g.subs(x, 0))
print("The two functions are equivalent")
```

```
When x is non-negative, f(x) - g(x) is 0, implying that f(x) and g(x) are equivalent.
f(0) - g(0) = 0
The two functions are equivalent
```

2a critical values of g

In [15]:
```
x = symbols('x')
g = -5 * x ** 4 - 20 * x ** 3 + 18
gprime = g.diff(x)
solve(gprime)
```

Out[15]: `[-3, 0]`

2b intervals where g is increasing/decreasing

(command below in case graphical method is chosen)

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

In [23]:
```
print("g'(-100) =", gprime.subs(x, -100))
print("g'(-1) =", gprime.subs(x, -1))
print("g'(1) =", gprime.subs(x, 1))
print("The function is increasing on (-oo, -3) and decreasing on (-3, 0)U(0, oo)")
```

```
g'(-100) = 19400000
g'(-1) = -40
g'(1) = -80
The function is increasing on (-oo, -3) and decreasing on (-3, 0)U(0, oo)
```

2c where g'' = 0

In [24]:
```
gdoubleprime = gprime.diff(x)
solve(gdoubleprime)
```

Out[24]:
```
[-2, 0]
```

2d intervals of concavity of g

(command below in case graphical method is chosen)

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

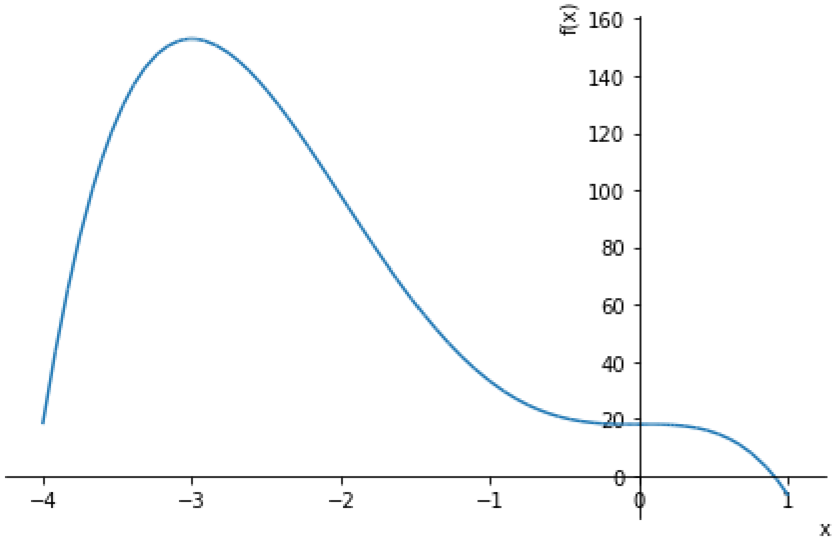
In [25]:
```
print("g'(-100) =", gdoubleprime.subs(x, -100))
print("g'(-1) =", gdoubleprime.subs(x, -1))
print("g'(1) =", gdoubleprime.subs(x, 1))
print("The function is concave down from (-oo, -2)U(0, oo) and concave up from (-2, 0)"
```

```
g'(-100) = -588000
g'(-1) = 60
g'(1) = -180
```

2e Plot to graphically confirm answers to (b) and (d)

In [27]:
```
matplotlib notebook
```

In [26]:
```
plot(g,(x,-4, 1))
```



Out[26]: <sympy.plotting.plot.Plot at 0x1fad4701a60>

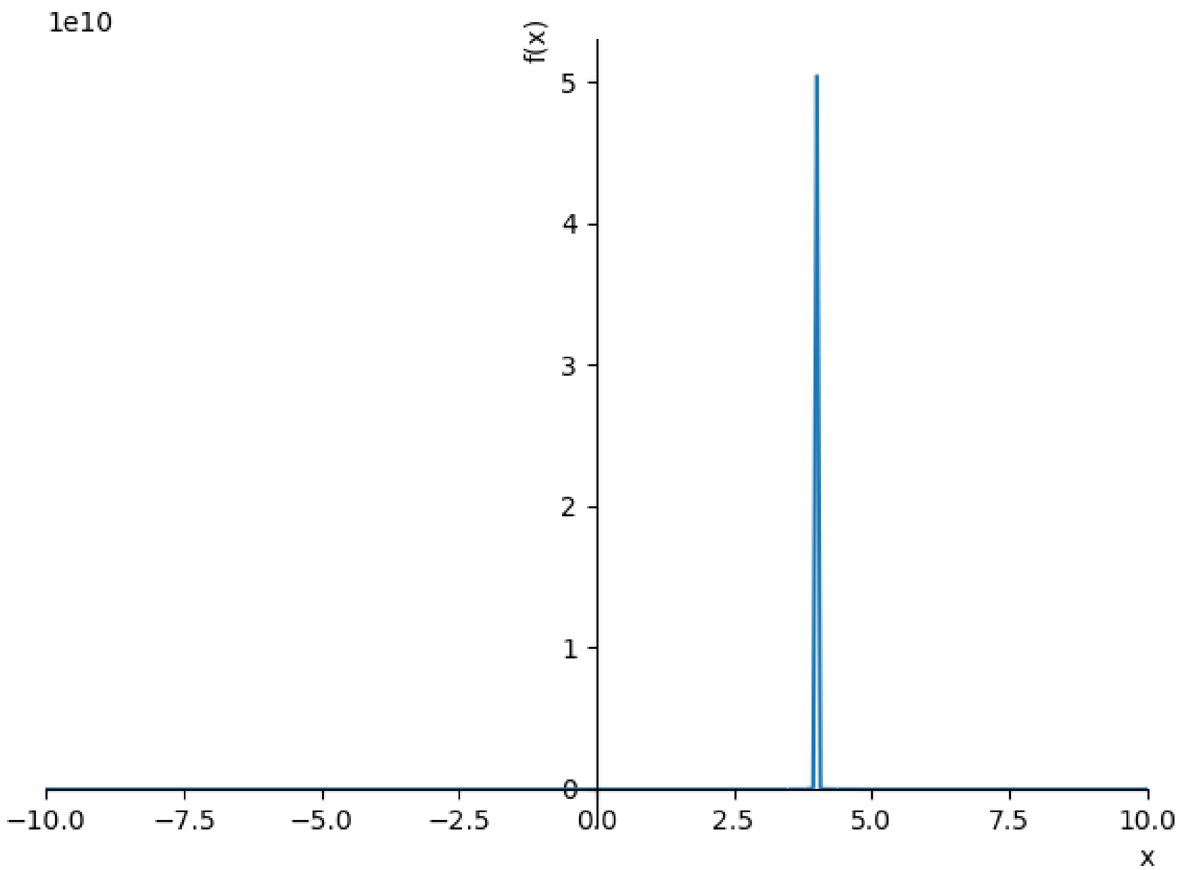
3a Plot of f and number of extrema and inflection points based on graph

In [28]:

```
matplotlib notebook
```

In [38]:

```
x = symbols('x', real=True)
f = (x ** 2 * (x + 1) ** 3)/((x - 2) ** 2 * (x - 4) ** 4)
plot(f, xlim=[-10,10])
print("There appears to be 3 local extrema and 4 inflection points")
```




```
There appears to be 3 local extrema and 4 inflection points
```

3b f' and critical values

In [62]:

```
dfdx = f.diff(x)
dfdx = simplify(dfdx)
sols = solve(dfdx)
print("Derivative: ", dfdx)
print("Critical Values:")
for sol in sols:
    print(re(sol.evalf()))
```

```
Derivative:  x*(x + 1)**2*(-2*x*(x - 4)*(x + 1) - 4*x*(x - 2)*(x + 1) + (x - 4)*(x - 2)*
(5*x + 2))/((x - 4)**5*(x - 2)**3)
Critical Values:
-1.00000000000000
0
-0.321983253601638
2.46674654943711
-20.1447632958355
```

3c intervals of increase or decrease

(command below in case graphical method is chosen)

In [ ]:

```
matplotlib notebook
```

In [69]:
```
print("f'(-100)\t=",  dfdx.subs(x, -100))
print("f'(-2)\t\t=",  dfdx.subs(x, -2))
print("f'(-1/2)\t=",  dfdx.subs(x, -1/2))
print("f'(-1/10)\t=", dfdx.subs(x, -1/10))
print("f'(1)\t\t=",   dfdx.subs(x, 1))
print("f'(100)\t\t=", dfdx.subs(x, 100))
print()
print("The function is decreasing on the interval (-oo, -20.14)U(-0.32, 0.00)U(2.47, oo
print("The function is increasing on the interval (-20.14, -1)U(-1,-0.32)U(0,2.47)")
```

```
f'(-100)        = -14456475/233492801152
f'(-2)          = 17/31104
f'(-1/2)        = 4.49795932191908e-5
f'(-1/10)       = -8.62207869908348e-5
f'(1)           = 164/243
f'(100)         = -9368853425/59954864062464

The function is decreasing on the interval (-oo, -20.14)U(-0.32, 0.00)U(2.47, oo)
The function is increasing on the interval (-20.14, -1)U(-1,-0.32)U(0,2.47)
```

3d intervals of concavity

(command below in case graphical method is chosen)

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

In [80]:
```
dfdx2 = dfdx.diff(x)
sols = solve(dfdx2)
print("The second derivative is", dfdx2)
print()
print("The x values of the inflection points are: ")
for sol in sols:
    print(re(sol.evalf()))

print()
print("f''(-100)\t=", dfdx2.subs(x, -100))
print("f''(-5)\t\t=", dfdx2.subs(x, -5))
print("f''(-2)\t\t=", dfdx2.subs(x, -2))
print("f''(-3/4)\t=", dfdx2.subs(x, -3/4))
print("f''(-1/2)\t=", dfdx2.subs(x, -1/2))
print("f''(0)\t\t=", dfdx2.subs(x, 0))
print()

print("The function is concave down on the interval (-oo, -35.31)U(-4.98,-1)U(-0.54,-0.
print("The function is concave up on the interval (-35.31,-4.98)U(-1,-0.54)U(-0.11,oo)"
```

```
The second derivative is x*(x + 1)**2*(-2*x*(x - 4) - 4*x*(x - 2) - 6*x*(x + 1) + (-4*x
- 4)*(x - 2) + (-2*x - 2)*(x - 4) + 5*(x - 4)*(x - 2) + (x - 4)*(5*x + 2) + (x - 2)*(5*x
+ 2))/((x - 4)**5*(x - 2)**3) + x*(2*x + 2)*(-2*x*(x - 4)*(x + 1) - 4*x*(x - 2)*(x + 1)
+ (x - 4)*(x - 2)*(5*x + 2))/((x - 4)**5*(x - 2)**3) - 3*x*(x + 1)**2*(-2*x*(x - 4)*(x +
1) - 4*x*(x - 2)*(x + 1) + (x - 4)*(x - 2)*(5*x + 2))/((x - 4)**5*(x - 2)**4) - 5*x*(x +
1)**2*(-2*x*(x - 4)*(x + 1) - 4*x*(x - 2)*(x + 1) + (x - 4)*(x - 2)*(5*x + 2))/((x - 4)*
*6*(x - 2)**3) + (x + 1)**2*(-2*x*(x - 4)*(x + 1) - 4*x*(x - 2)*(x + 1) + (x - 4)*(x -
2)*(5*x + 2))/((x - 4)**5*(x - 2)**3)

The x values of the inflection points are:
-1.00000000000000
-35.3113256040630
```

```
-4.97777332154410
-0.542718018300141
-0.108385637343953

f''(-100)       = -6122901477/6605044358987776
f''(-5)         = 6568/1275989841
f''(-2)         = -343/373248
f''(-3/4)       = 0.000154684231019964
f''(-1/2)       = -7.16301527356753e-5
f''(0)          = 1/512
```

3e actual number of local extrema and inflection points

In [ ]:

```
print("The are actually 4 local extrema and 5 inflection points")
```

4a Define f and g for ln(y)

In [81]:

```
x = symbols('x', real=True)
y = (1 - 6 * x) ** (1 / x)
f = ln(1 - 6 * x)
g = x
print("ln(y) = ln(1 - 6x)/x")
print("f(x) = ln(1-6x)")
print("g(x) = x")
```

```
ln(y) = ln(1 - 6x)/x
f(x) = ln(1-6x)
g(x) = x
```

4b limits of f and g as x approaches 0

In [84]:

```
print("limit of f =", limit(f, x, 0))
print("limit of g =", limit(g, x, 0))
```

```
limit of f = 0
limit of g = 0
```

4c Apply L'Hospital's Rule if applicable

In [87]:

```
l = limit(f.diff(x)/g.diff(x), x, 0)
print(l)
print("Limit of y as x approaches 0 is", exp(l))
```

```
-6
Limit of y as x approaches 0 is exp(-6)
```

4d limit directly in Python

In [88]:
```
print(limit(y, x, 0))
```

```
exp(-6)
```

In [ ]: