Name	UIN
Huy Lai	132000359
Alex Nuccitelli	000000000
Cole Jahnke	530009075

05 October 2021

MATH 151-557

In [2]:

```
from sympy import *
from sympy.plotting import (plot, plot_parametric)
```

1a Tangent line at x=2

In [10]:

```
x = symbols('x')
f = 5 * x / (1 + x**2)
m = f.diff().subs(x, 2)
print(m)
```

```
-3/5
```

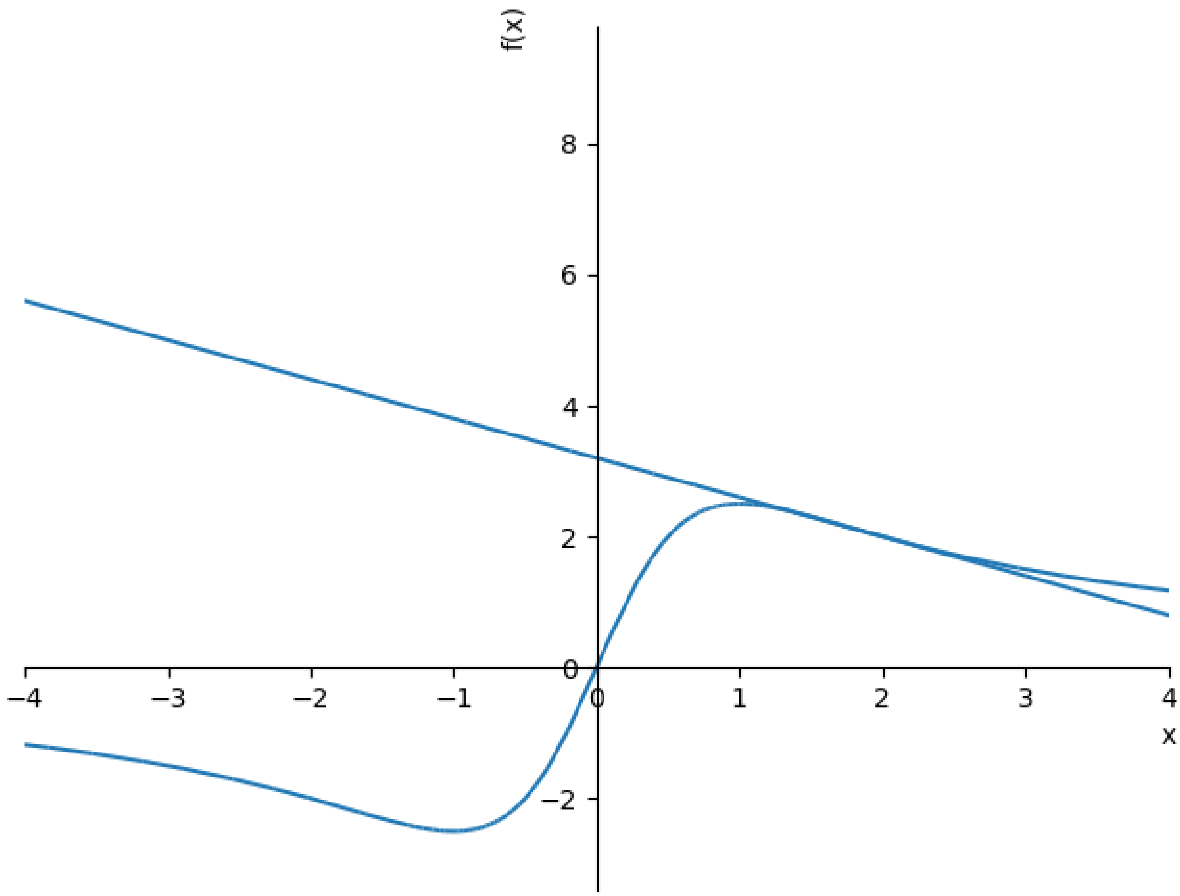
1b Graph of f and tangent line

In [4]:

```
matplotlib notebook
```

In [13]:

```
tan_line = m * (x - 2) + f.subs(x, 2)
plot(f, tan_line, xlim=[-4,4])
```



Out[13]: <sympy.plotting.plot.Plot at 0x21d81f043d0>

2a Veclocity after 1 second

In [16]:

```
t = symbols('t')
s = 1 + 10 * t - 1.86 * t ** 2
v = s.diff()
v.subs(t, 1)
```

Out[16]: $\displaystyle 6.28$

2b when v=0 and corresponding height

In [17]:

```
solve(v, t)
```

Out[17]: [2.68817204301075]

2c velocity when hits the surface

In [20]:

```
v.subs(t, solve(s, t)[1])
```

Out[20]:

$\displaystyle -10.3653268158799$

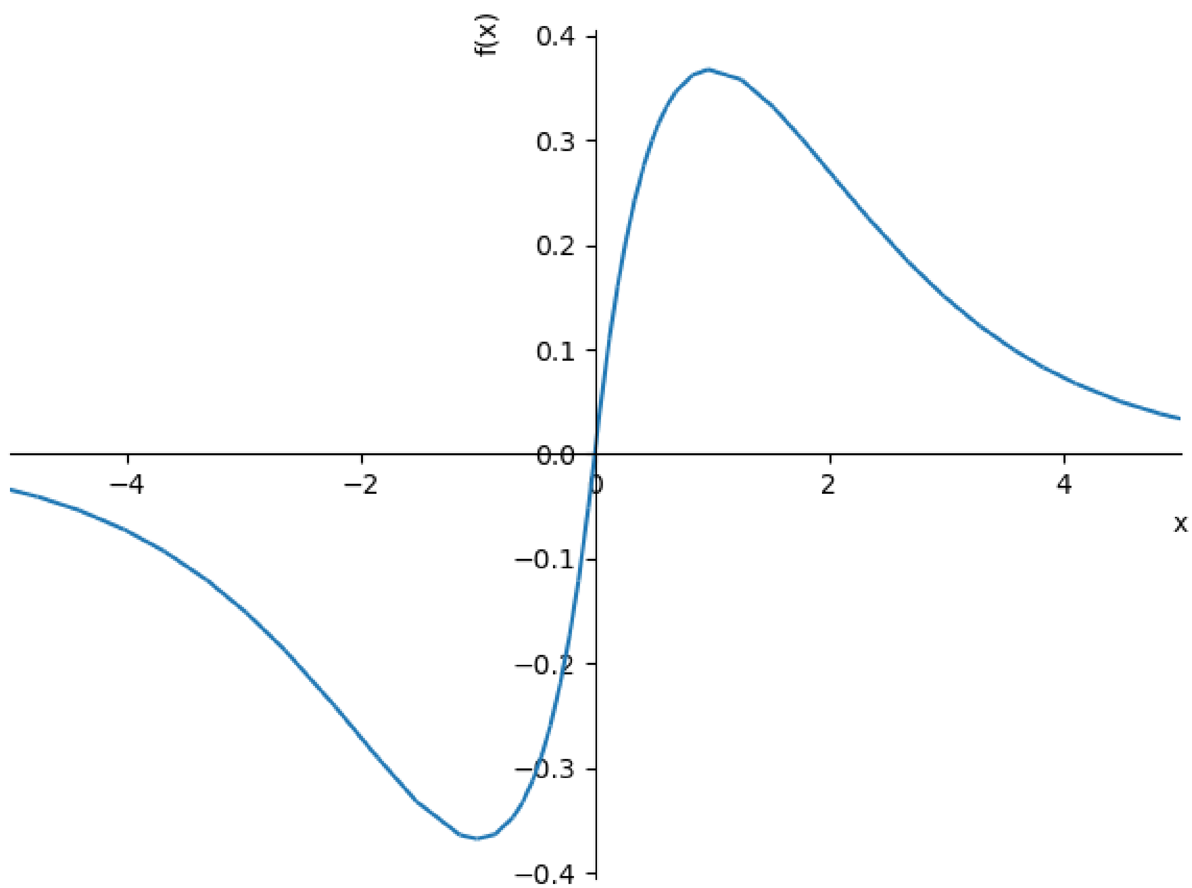
3a Plot of f

In [6]:

```
matplotlib notebook
```

In [30]:

```
x = symbols('x', real = True)
f = x*exp(-abs(x))
plot(f, xlim=[-5,5])
```



Out[30]: <sympy.plotting.plot.Plot at 0x21d82cc6ac0>

3b Equations of horizontal tangent lines

In [32]:

```
f_pos = x * exp(-x)
f_neg = x * exp(x)
print(solve(f_pos.diff(), x), solve(f_neg.diff(), x))
```

```
[1] [-1]
```

4a first 16 derivatives

In [33]:

```
x = symbols('x')
f = x * sin(x)
diffs = [f.diff(x,n) for n in range(1,17)]
print(diffs)
```

```
[x*cos(x) + sin(x), -x*sin(x) + 2*cos(x), -(x*cos(x) + 3*sin(x)), x*sin(x) - 4*cos(x), x
*cos(x) + 5*sin(x), -x*sin(x) + 6*cos(x), -(x*cos(x) + 7*sin(x)), x*sin(x) - 8*cos(x), x
*cos(x) + 9*sin(x), -x*sin(x) + 10*cos(x), -(x*cos(x) + 11*sin(x)), x*sin(x) - 12*cos
(x), x*cos(x) + 13*sin(x), -x*sin(x) + 14*cos(x), -(x*cos(x) + 15*sin(x)), x*sin(x) - 16
*cos(x)]
```

4b formula for nth, (n+1)th, (n+2)th, and (n+3)th derivative if n divisible by 4

In [35]:

```
print("f^(n)(x) = x * sin(x) - n * cos(x) ")
print("f^(n + 1)(x) = x * cos(x) + (n + 1) * sin(x)")
print("f^(n + 2)(x) = -x * sin(x) + (n + 2) * cos(x) ")
print("f^(n + 3)(x) = -1(x * cos(x) + (n + 3) * sin(x)) ")
```

```
f^(n)(x) = x * sin(x) - n * cos(x)
f^(n + 1)(x) = x * cos(x) + (n + 1) * sin(x)
f^(n + 2)(x) = -x * sin(x) + (n + 2) * cos(x)
f^(n + 3)(x) = -1(x * cos(x) + (n + 3) * sin(x))
```

In [ ]: