transform

June 28, 2021

[1]:

```
import numpy as np
import matplotlib.pyplot as plt
import sage.plot.scatter_plot as scatter
%matplotlib inline
```

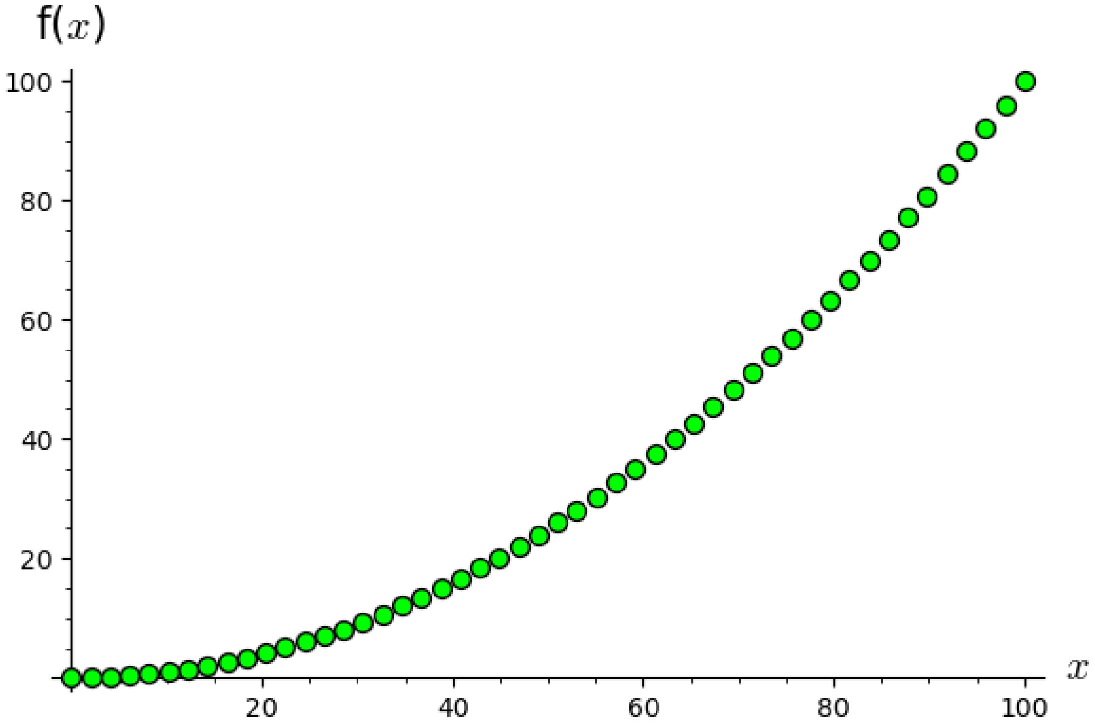
[2]:

```
def plotTangent(equation):
    deq = equation.derivative(x)

    x_data = np.linspace(0, 100, num=50)

    # Makes the equation callable
    func = fast_callable(equation, vars=[x])
    slope = fast_callable(deq, vars=[x])

    # Plotting the function
    plots = [ (x_data[i], func(x_data[i])) for i in range(len(x_data)) ]

    g = Graphics()
    g += scatter.scatter_plot(plots, facecolor='lime')

    # Plotting the tangent lines
    p = Graphics()
    for i in range(0, len(x_data), 5):
        x0 = x_data[i]
        y0 = func(x_data[i])
        s = slope(x_data[i])

        points = [ (x, y0+s*(x-x0)) for x in [0, x0-y0/s, x0, x0*10] ]
        p += line(points)


    g.save("Plotted.png", axes_labels=['$x$','f($x$)'])
    p.save("Tangents.png", axes_labels=['$x$','f($x$)'], xmin=0, xmax=100,␣
↪ymin=-1, ymax=func(x_data[-1]))

    return p, g
```

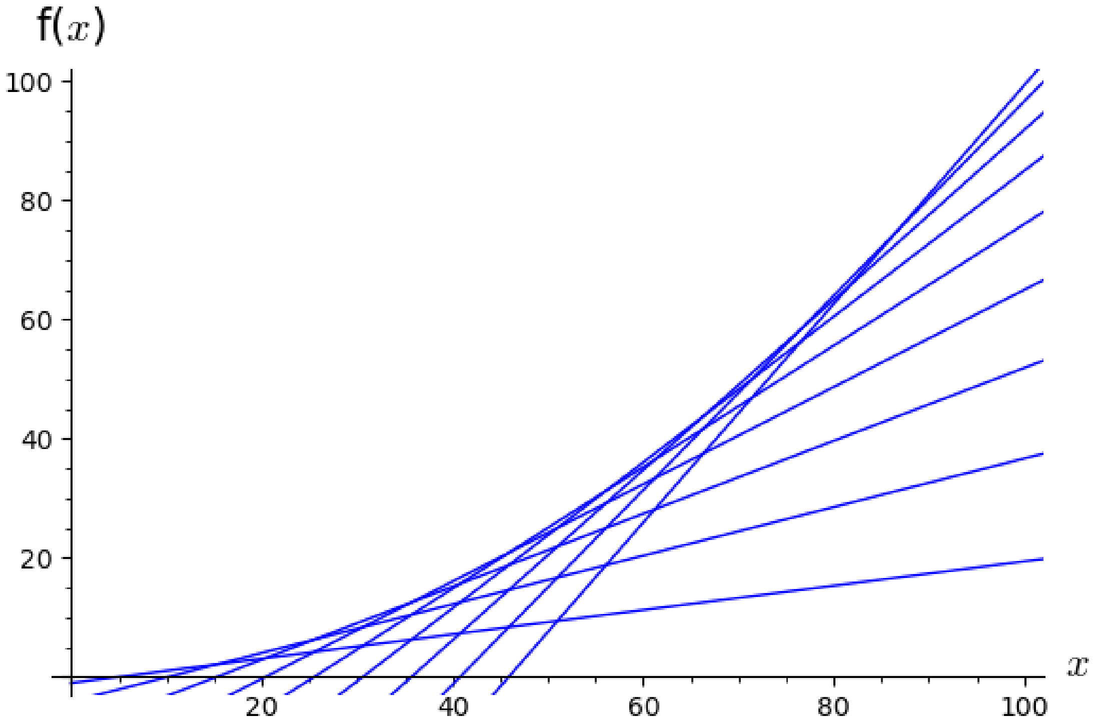
```
[3]: p, g = plotTangent(0.01*x^2)
```

```
<ipython-input-2-56a07317fc48>:20: RuntimeWarning: invalid value encountered in
double_scalars
  points = [ (x, y0+s*(x-x0)) for x in [Integer(0), x0-y0/s, x0, x0*Integer(10)]
]
```

```
[4]: g.show()
```



```
[6]: p.show(xmin=0, xmax=100, ymin=-1, ymax=100)
```


f(x)
100
80
60
40
20
x
20
40
60
80
100

[ ]: