Assignment02

March 21, 2019

[Taylor Approximation]

1. Define a differentiable function that maps from real number to real number.
2. Define a domain of the function.
3. Plot the function.
4. Select a point within the domain.
5. Mark the selected point on the function.
6. Define the first-order Taylor approximation at the selected point.
7. Plot the Taylor approximation with the same domain of the original function.

1 First-order Taylor approximation

suppose $f : R^n \to R$
first-order Taylor approximation of f, near point z:

$$\hat{f}(x) = f(z) + \frac{\partial f}{\partial x_1}(z)(x_1 - z_1) + ... + \frac{\partial f}{\partial x_n}(z)(x_n - z_n)$$

$\hat{f}(x)$ is very close to $f(x)$ when x_i are all near z_i
$\hat{f}$ is an affine function of x
can write using inner product as

$$\hat{f}(x) = f(z) + \nabla f(z)^T (x - z)$$

where n-vector $\nabla f(z)$ is the gradient of f at z,

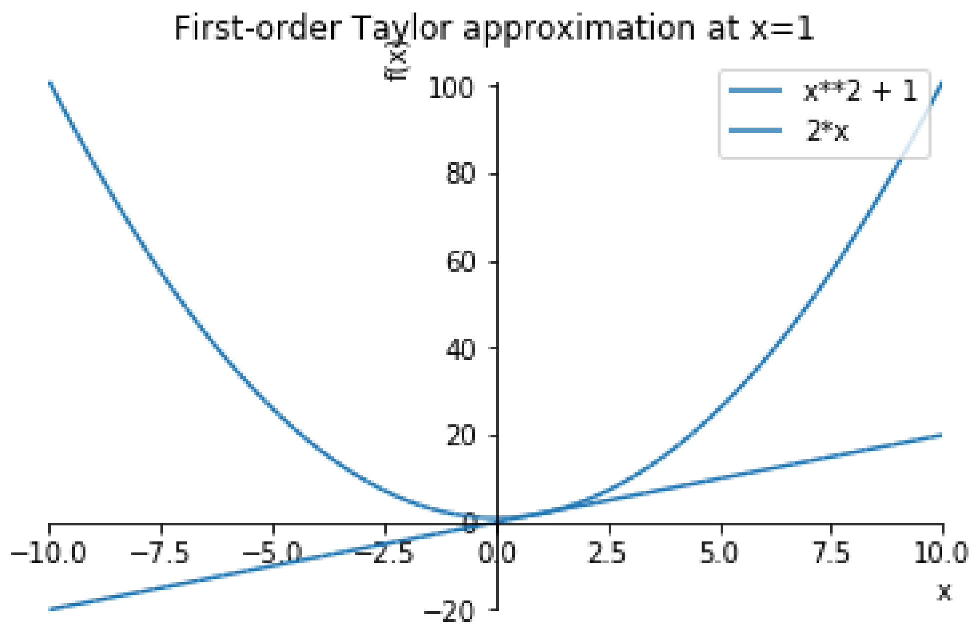
$$\nabla f(z) = (\frac{\partial f}{\partial x_1}(z), ..., \frac{\partial f}{\partial x_n}(z))$$

First-order Taylor Approximation of $f(x) = x^2 + 1$ at $x = 1$

In [3]:
```python
import sympy as sy
from sympy import *
from sympy.plotting import plot
import numpy as np
import matplotlib.pyplot as plt

x = symbols('x')
f = x**2 + 1
```

```python
# Taylor approximation at x=1
fhat = f.subs({x:1}) + (f.diff(x)).subs({x:1})*(x-1)
p=plot(f, fhat, legend=True, title='First-order Taylor approximation at x=1')
```




```
In [ ]:
```