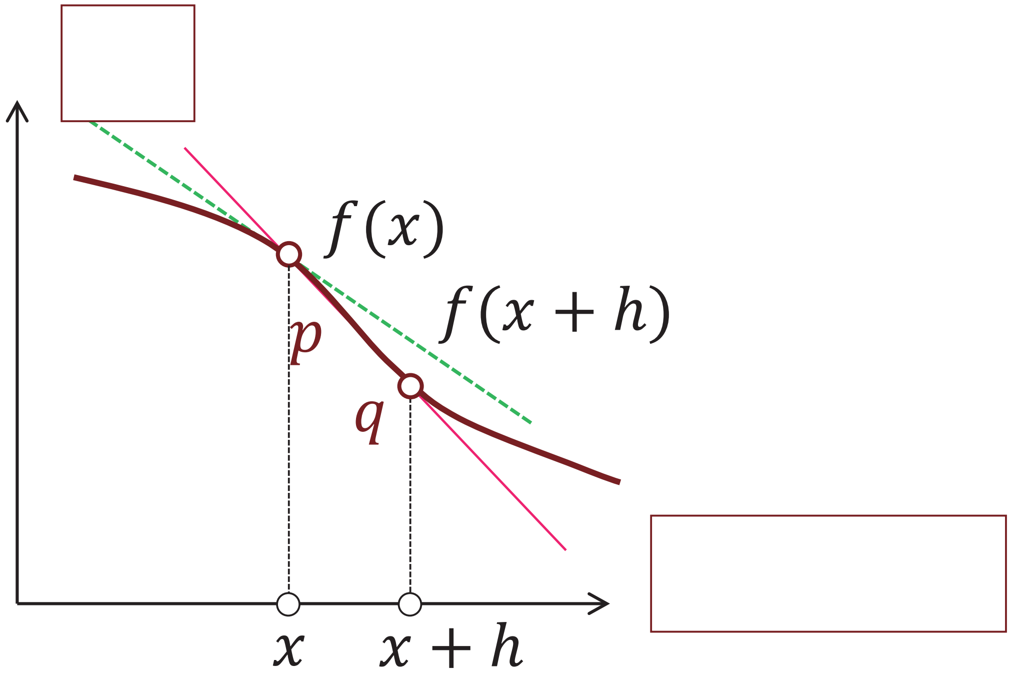
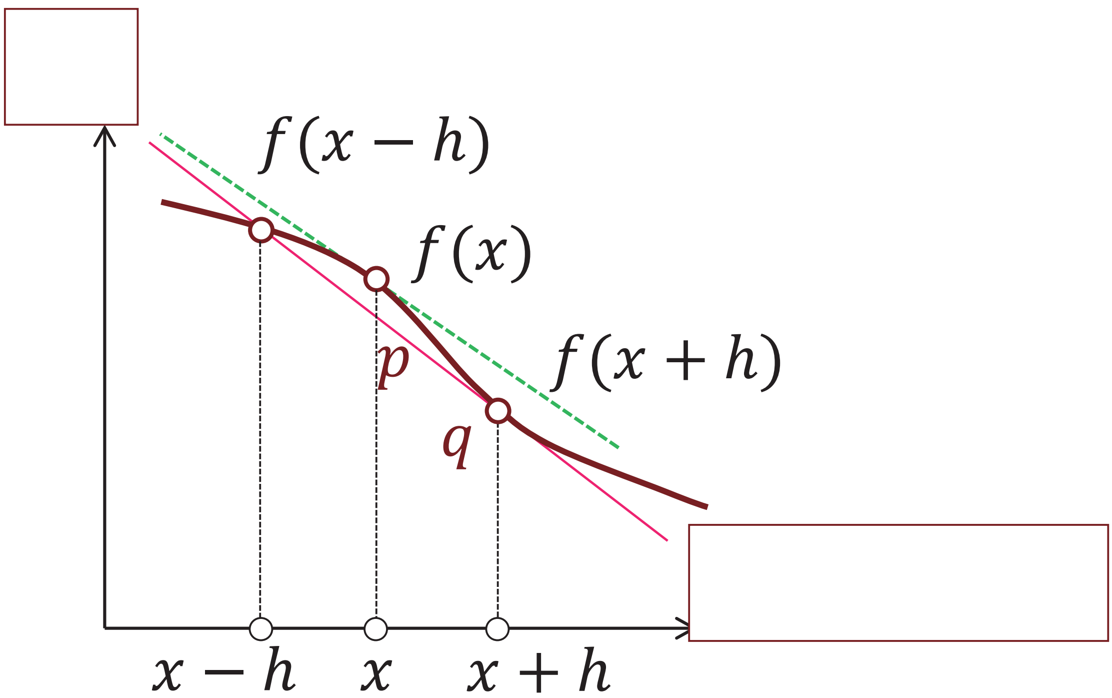
数値微分



数値微分の工夫

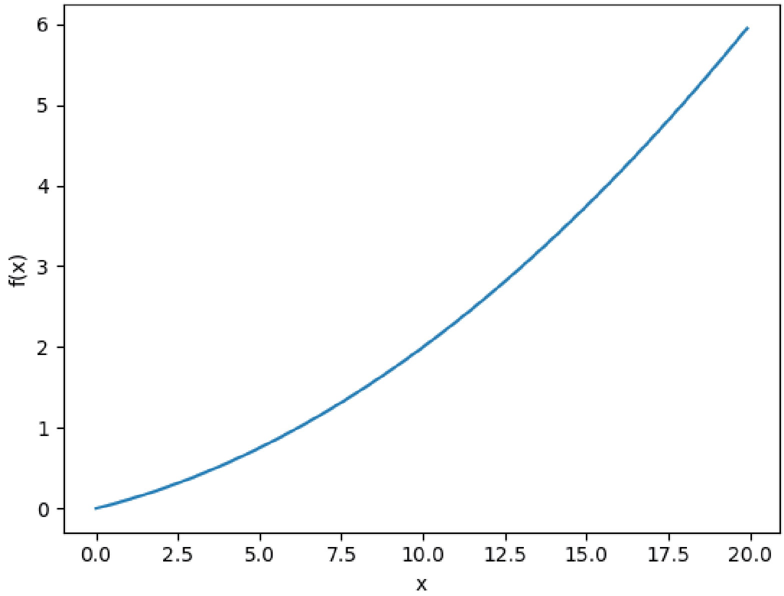


数値微分の例

P100

$$f(x) = \boxed{\quad}$$

```
import numpy as np
import matplotlib.pylab as plt

def function_1(x):
   return 0.01*x**2 + 0.1*x

x = np.arange(0.0, 20.0, 0.1)
y = function_1(x)

plt.xlabel("x")
plt.ylabel("f(x)")
plt.plot(x, y)
plt.show()
```



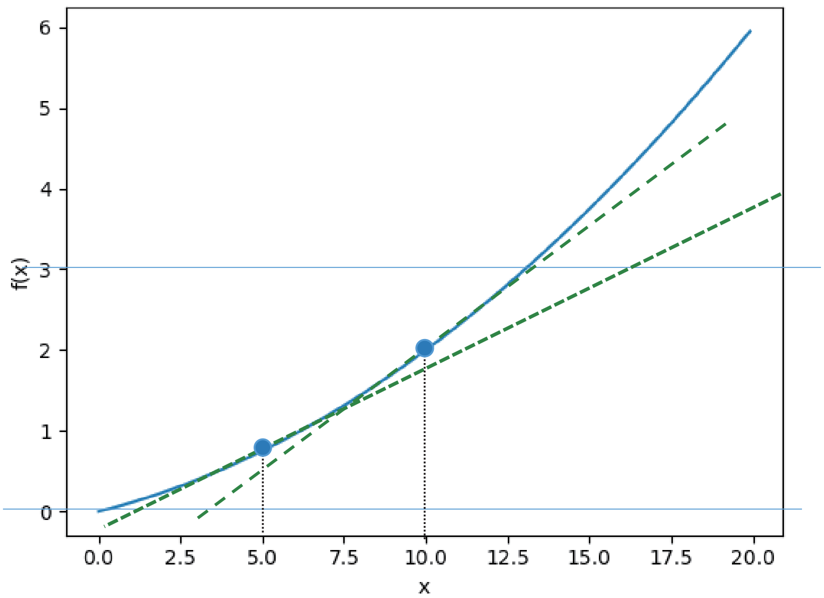
解析による傾きの算出

P100

$$f(x) = \boxed{\quad} \Rightarrow \frac{df(x)}{dx} = \boxed{\quad}$$

$$\left.\frac{df(x)}{dx}\right|_{x=5} = \boxed{\quad}$$

$$\left.\frac{df(x)}{dx}\right|_{x=10} = \boxed{\quad}$$



Pythonでの計算結果 P100

```
import numpy as np

def function_1(x):
    return 0.01*x**2 + 0.1*x

def numerical_diff_A(f, x):
    h = 10e-50
    return (f(x+h) - f(x)) / h

def numerical_diff_B(f, x):
    h = 1e-4
    return (f(x+h) - f(x-h)) / (2*h)

numerical_diff_A(function_1, 5)

numerical_diff_A(function_1, 10)

numerical_diff_B(function_1, 5)

numerical_diff_B(function_1, 10)
```

```
>>> numerical_diff_A(function_1, 5)
0.0
>>>
>>> numerical_diff_A(function_1, 10)
0.0



>>>
>>> numerical_diff_B(function_1, 5)
0.1999999999990898
>>>
>>> numerical_diff_B(function_1, 10)
0.2999999999986347
```


偏微分（2変数を例として） P103

■ $\boxed{}$する時の偏微分

$$\frac{\partial f(x_0, x_1)}{\partial x_0} = \lim_{h \to 0} \frac{\boxed{}}{h}$$

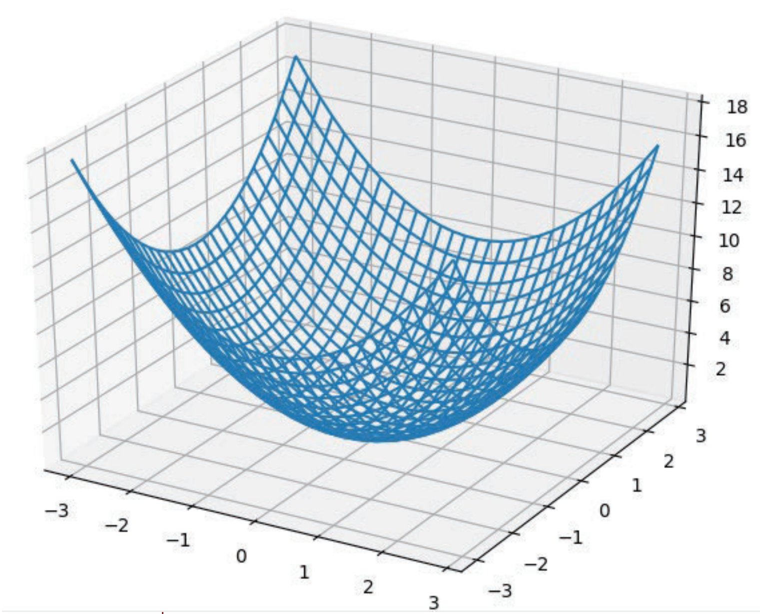
■ $\boxed{}$する時の偏微分

$$\frac{\partial f(x_0, x_1)}{\partial x_1} = \lim_{h \to 0} \frac{\boxed{}}{h}$$

偏微分の例 P103

$$f(x_0, x_1) = \boxed{}$$

```
import numpy as np
import matplotlib.pylab as plt
from mpl_toolkits.mplot3d import Axes3D

def function_2(x0, x1):
   return x0**2 + x1**2

x0 = np.arange(-3.0, 3.0, 0.2)
x1= np.arange(-3.0, 3.0, 0.2)

x0, x1 = np.meshgrid(x0, x1)

y = function_2(x0, x1)

fig = plt.figure()
ax = Axes3D(fig)
ax.plot_wireframe(x0, x1, y)
plt.show()
```



偏微分の例 P103

$$\frac{\partial f(x_0,x_1)}{\partial x_0} = \boxed{}$$

$$\left.\frac{\partial f(x_0,x_1)}{\partial x_0}\right|_{\substack{x_0=3\\ x_1=4}} = \boxed{}$$

$$\frac{\partial f(x_0,x_1)}{\partial x_1} = \boxed{}$$

$$\left.\frac{\partial f(x_0,x_1)}{\partial x_1}\right|_{\substack{x_0=3\\ x_1=4}} = \boxed{}$$

Pythonでの計算結果

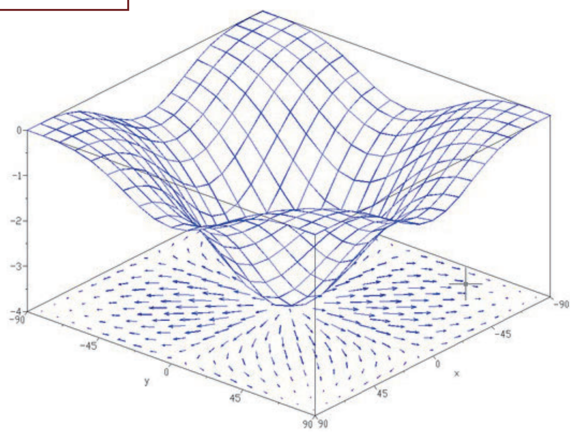

```
def function_2(x0, x1):
    return x0**2 + x1**2

def numerical_diff_x0(f, x0, x1):
    h = 1e-4
    return (f(x0+h,x1) - f(x0-h,x1)) / (2*h)

def numerical_diff_x1(f, x0, x1):
    h = 1e-4
    return (f(x0,x1+h) - f(x0,x1-h)) / (2*h)

numerical_diff_x0(function_2, 3, 4)

numerical_diff_x1(function_2, 3, 4)
```

```
>>> numerical_diff_x0(function_2, 3, 4)
6.00000000000378
>>>

>>> numerical_diff_x1(function_2, 3, 4)
7.999999999999119
>>>
```


勾配（3次元での例）

■勾配　局面上の任意の点において，　　　　　　　を
　　　　指すベクトル．　　　　　とも呼ぶ．

□ ＝ □ ＝ □



Wikipediaより

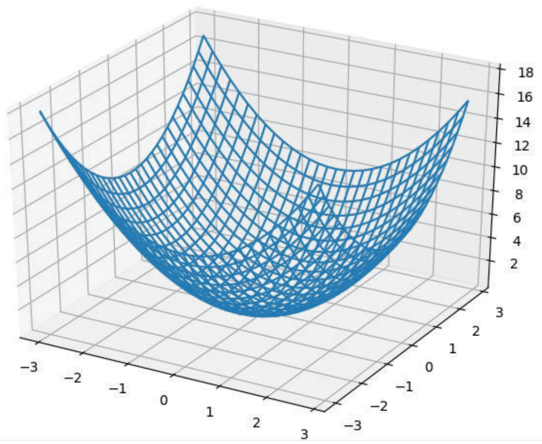
勾配の例

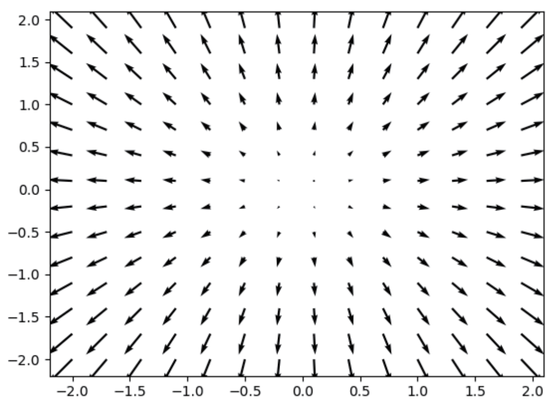
$$f(x_0, x_1) = \boxed{}$$

$$\nabla f = \mathrm{grad} f = \left(\frac{\partial f}{\partial x_0}, \frac{\partial f}{\partial x_1}\right) = \boxed{}$$

勾配の計算結果


```
import matplotlib.pyplot as plt
import numpy as np

X0,X1 = np.meshgrid(np.arange(-2.0, 2.0,0.3),
np.arange(-2.0, 2.0,0.3))
U = 2*X0
V = 2*X1

plt.figure()
Q = plt.quiver(X0, X1, U, V, units='width')

plt.show()
```

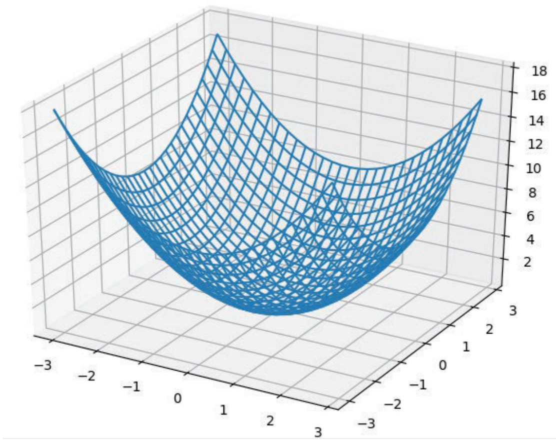


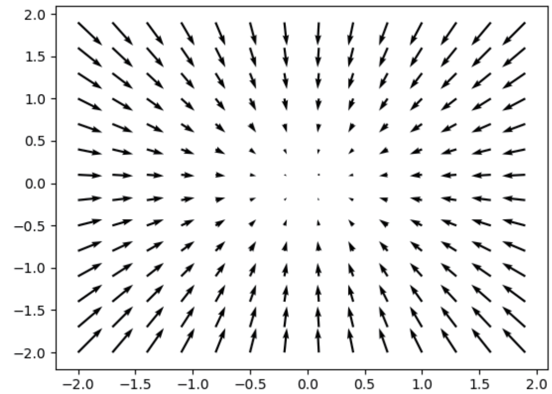


勾配により最小値に行き着くには？ P105

■勾配とは　　　に行く

$$-\nabla f = -\left(\frac{\partial f}{\partial x_0}, \frac{\partial f}{\partial x_1}\right)$$





勾配法 P107

■勾配と　　　に行くことにより関数の　　　を　　　により求める.

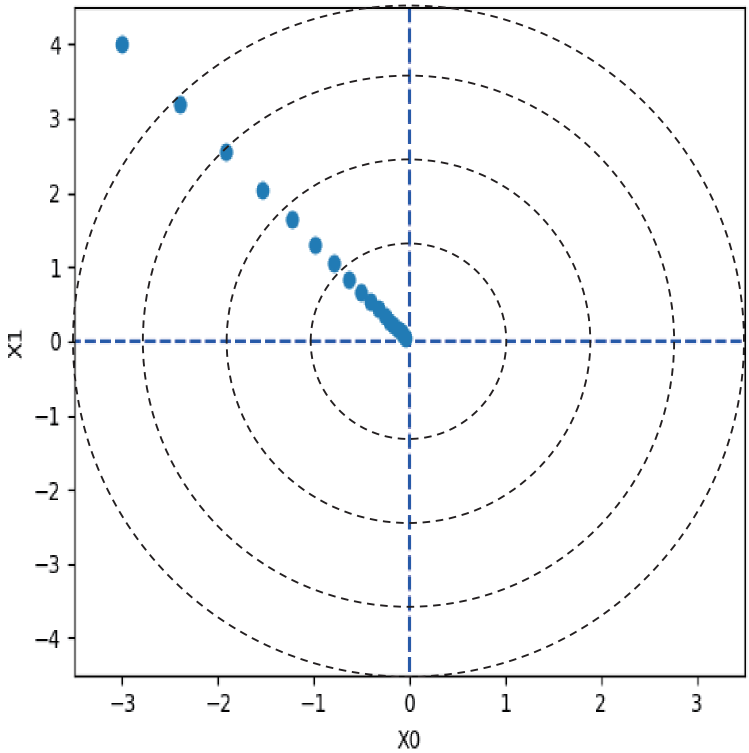
$\leftarrow$

$f(x_0, x_1) =$ 　　　の場合には

$$\begin{cases} \quad \leftarrow \quad = \quad \\ \quad \leftarrow \quad = \quad \end{cases}$$

勾配下降法による最小値算出の例 P109

$$f(x_0, x_1) = \boxed{} \qquad \eta = \boxed{}$$



勾配法による重みのアップデート P110

$$W = \begin{pmatrix} w_{11} & \cdots & w_{1n} \\ \vdots & \ddots & \vdots \\ w_{m1} & \cdots & w_{mn} \end{pmatrix}$$

$$\frac{\partial E}{\partial W} = \boxed{}$$

$$\boxed{ \leftarrow }$$