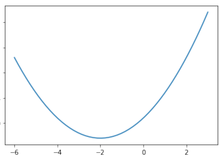
第三阶段-笔记

一元方程

```
def f(x):
    return x ** 2 + 4*x + 1

def df(x):
    return 2 * x + 4

x_old = 3.14

for itr in range(20):
    x_new = x_old -  0.1 * df(x_old)
    x_old = x_new
    print("f({})={}".format(x_new,f(x_new)))
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(-6,3)
plt.plot(x,f(x))
plt.show()
```

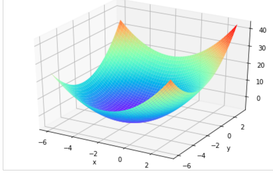
```
f(2.112)=13.908544000000001
f(1.2896)=7.82146816
f(0.63168)=3.9257396224
f(0.105344)=1.4324733583359999
f(-0.3157248)=-0.16321705066496017
f(-0.65257984)=-1.1844589124255744
f(-0.9220638720000001)=-1.8380537033952368
f(-1.1376510976)=-2.2563543705295155
f(-1.31012087808)=-2.5240667971388895
f(-1.448096702464)=-2.6954027501688893
f(-1.5584773619712)=-2.805057760108089
f(-1.646781889576696)=-2.8752369664691773
f(-1.7174255511661568)=-2.920151658540273
f(-1.7739404093292543)=-2.948897061465775
f(-1.8191523276634034)=-2.9672941193380957
f(-1.8553218619707228)=-2.979068236376381 4
f(-1.8842574895765782)=-2.986603671280884
f(-1.9074059916612627)=-2.9914263496197657
f(-1.9259247933290102)=-2.994512863756 65
f(-1.9407398346632 08)=-2.996488232804256
```



二元方程

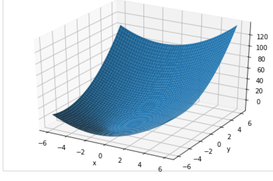
```
def f_2d(x,y):
    return x ** 2 + 4*x + y ** 2 + 4*y + 1

def df_2d(x,y):
    return 2 * x + 4, 2 * y +4

x,y = 4,4
for itr in range(20):
    v_x,v_y = df_2d(x,y)
    x, y = x - 0.1 * v_x, y - 0.1 * v_y
    print("f({},{})={}".format(x,y,f_2d(x,y)))


from mpl_toolkits.mplot3d import Axes3D
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(-6,3)
y = np.linspace(-6,3)

X,Y = np.meshgrid(x,y)
fig = plt.figure()
ax = Axes3D(fig)
surf = ax.plot_surface(X, Y, f_2d(X,Y),rstride=1, cstride=1, cmap='rainbow')
plt.xlabel('x')
plt.ylabel('y')

plt.show()
```

```
f(2.82,8)=39.08
f(1.8399999999999991,8.399999999999999)=22.4912
f(1.071999999999999 81,0.7199999999999998)=11.874367999999999
f(0.45759999999999980,0.4575999999999998)=5.079595519999998
f(-0.033920000000000 17,-0.03392000000000017)=0.7309411327999986
f(-0.42713600000000002,-0.42713600000000002)=-2.052197675008001
f(-0.741708800000000 02,-0.74170880000000002)=-3.833406512005121
f(-0.99336704000000001,-0.99336704000000001)=-4.971380167683278
f(-1.1946936320000001,-1.1946936320000001)=-5.702963307317297
f(-1.3557549056,-1.3557549056)=-6.16989651668307
f(-1.48460392448,-1.48460392448)=-6.468733770677165
f(-1.587683139584,-1.587683139584)=-6.659989613233385
f(-1.6701465116672,-1.6701465116672)=-6.782293525693665
f(-1.736117209333376,-1.736117209333376)=-6.860731745580395
f(-1.788893767467008,-1.788893767467008)=-6.910868131714524
f(-1.831115013973606 3,-1.831115013973606 3)=-6.942955728979 73
f(-1.864892011178885 1,-1.864892011178885 1)=-6.963491662713427
f(-1.891913608943108 1,-1.891913608943108 1)=-6.976634664136593 5
f(-1.913530887154486 5,-1.913530887154486 5)=-6.985046180474 42
f(-1.930824709723589,-1.930824709723589)=-6.990429558430349
```



练习

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

def f_2d(x,y):
    return x ** 2 + 3 * x + y ** 2 + 8 * y + 1


def df_2d(x,y):
    return 2 * x + 3, 2 * y + 8

x,y = 4,4
for itr in range(200):
    v_x,v_y = df_2d(x,y)
    x, y = x - 0.1 * v_x, y - 0.1 * v_y
    # print("f({},{})={}".format(x,y,f_2d(x,y)))


x = np.linspace(-6,6)
y = np.linspace(-6,6)

X,Y = np.meshgrid(x,y)
fig = plt.figure()
ax = Axes3D(fig)
surf = ax.plot_surface(X, Y, f_2d(X,Y),rstride=1, cstride=1)
plt.xlabel('x')
plt.ylabel('y')

plt.show()
```



```

```