15091060

5.9

γ=0.9,ρ=0.01

:(1.2,1.2)

995 : [1.05604686 1.11578718] : 0.00179701029991 996 : [1.05626461 1.11558871] : 0.00179701029991 997 : [1.05598175 1.11562688] : 0.00199667811102 998 : [1.0562047 1.11541547] : 0.00179701029991 999 : [1.05588661 1.11547042]

:(-1.2,1)

995 : [1.32340857 1.75425742] : 0.00117901845777 996 : [1.32442297 1.75358604] : 0.00131002050864 997 : [1.32321891 1.75371971] : 0.00117901845777 998 : [1.32421119 1.75305677] : 0.00131002050864 999 : [1.32302971 1.75318213]

γ=0.9,ρ=0.01

:(1.2,1.2)

1 : [1.19591837 1.43020408] : 0.6561 2 : [1.0678032 1.12378445] : 1 3 : [1.05197555 1.10640204] : 1 4 : [1.00247988 1.00251608] : 1 5 : [1.00081549 1.00162887] : 1 6 : [1.00000045 1.00000024] : 1 7 : [1. 1.]

:(-1.2,1)

15 : [0.90650146 0.82113373] : 1 16 : [0.98981611 0.97279461] : 1 17 : [0.99408025 0.98817735] : 1 18 : [0.99997855 0.99992231] : 1 19 : [0.99999985 0.9999997] : 1 20 : [1. 1.]

```
#-*- coding: UTF-8 -*-
"""
Newton
Rosenbrock
  f(x)=100*(x(2)-x(1).^2).^2+(1-x(1)).^2
  g(x)=(-400*(x(2)-x(1)^2)*x(1)-2*(1-x(1)),200*(x(2)-x(1)^2))^(T)
"""
rou=0.01
import numpy as np
import matplotlib.pyplot as plt
def F(x):
    return 100*(x[1]-x[0]**2)**2+(1-x[0])**2;

def jacobian(x):
    return np.array([-400*x[0]*(x[1]-x[0]**2)-2*(1-x[0]),200*(x[1]-x[0]**2)])

def hessian(x):
    return np.array([[-400*(x[1]-3*x[0]**2)+2,-400*x[0]],[-400*x[0],200]])

def phi(a,x,p):
    return F(x+a*p)


def Apha(x,p,g):
    a=1
    while(phi(a,x,p)>=F(x)+rou*np.dot(p.transpose(),g)*a):
        a=0.9*a
    return a

X1=np.arange(-1.5,2+0.05,0.05)
X2=np.arange(-3.5,3+0.05,0.05)
[x1,x2]=np.meshgrid(X1,X2)
f=100*(x2-x1**2)**2+(1-x1)**2; #
plt.contour(x1,x2,f,20) #    20


def newton(x0):

    print('  :')
    print(x0,'\n')
    W=np.zeros((2,10**3))
    i = 1
    imax = 1000
    W[:,0] = x0
    x = x0
    delta = 1
```

```
    alpha = 1

    while i<imax and delta>10**(-10):
        p = -np.dot(np.linalg.inv(hessian(x)),jacobian(x))
        #p=-jacobian(x)
        x0 = x
       # alpha=1.0*np.dot(jacobian(x).transpose(),jacobian(x))/np.dot((np.dot(jacobian(x).t
        alpha=Apha(x,p,jacobian(x))
        print('  :')
        print(alpha)
       # alpha=1.0
        x = x + alpha*p
        W[:,i] = x
        delta = sum((x-x0)**2)
        print('  %d    :'%(i))
        print(x)
        i=i+1
    W=W[:,0:i]  #
    return W

x0 = np.array([-1.2,1])
W=newton(x0)

plt.plot(W[0,:],W[1,:],'g*',W[0,:],W[1,:]) #
plt.show()
```

5.19

```
N=8;
G=ones(N,N);
b=ones(N,1);
x0=zeros(N,1);
for i= 1:N
    for j =1:N
         G(i,j) = 1/(i+j-1);
    end
end
    x = x0;
    g = G*x-b;
    p = -g;
    k = 0;
    while 1
        if norm(g, 2)<1e-6
            break
        end
```

```
        k = k + 1;

        d=G*p;
        a=(g'*g)/(p'*d);
        x = x+a*p;
        t=g+a*d;
       beta=(t'*t)/(g'*g);
       g=t;
       p=-g+beta*p;
    end
k
x
```

n=5	n=8	n=12	n=20
k=6	k=19	k=35	k=66
x	x	x	x
5.00000002066037	5.89856766292802e-11	-9.60895537135449	-10.9749126510893
-120.000000009110	-6.97176032007019e-11	815.396945500909	1050.92924583463
629.999999984204	-5.15860286852761e-10	-16496.5601423969	-23956.2795083182
-1120.00000001358	1.12339461172630e-09	135510.323445556	220425.673599387
629.999999984357	3.17065483551747e-10	-536481.215700387	-965346.669654001
	-6.55028164772178e-10	1025399.39571366	1990103.06862559
	-6.56530674556942e-10	-642578.292665479	-1252700.00760310
	4.58964570698870e-10	-657590.976689034	-1343474.30912827
		804243.884701944	883233.066644487
		663072.549158718	1687963.95092874
		-1241279.51532581	388212.758339880
		465506.443773192	-1305525.72041703
			-1710545.71991628
			-528251.066590642
			1208686.49124660
			2002890.86386703
			944594.155587335
			-1434053.34920119
			-2650953.30867554
			1887855.72260968

5.21

```
N=4;
G=zeros(N,N);
b=[-1,0,2,5^(0.5)]
x0=zeros(N,1);
for i= 1:N
```

```
     G(i,i) = 2;
end

for i= 1:N-1
    G(i+1,i)=-1;
    G(i,i+1)=-1;
end
    x = x0;
    g = G*x-b;
    p = -g;
    k = 0;
    while 1
        if norm(g, 2)<1e-6
            break
        end
        k = k + 1;

        d=G*p;
        a=(g'*g)/(p'*d);

        x = x+a*p;
        t=g+a*d;
       beta=(t'*t)/(g'*g);
       g=t;
       p=-g+beta*p;

    end
```

k = 3

g=[-2 ,-1 , 1.7639 , 2.4721]

Gg=[-3 ,-1.7639 , 2.0557 , 3.1803]

GGg=[-4.2361, -2.5836 , 2.6950 , 4.3050]