Homework 9

1 Question 50

Minimum at $(x,\ y)\ =\ (1.0,\ 1.0)$
$F(x=1,\ y=1)\ \approx\ 2.389604\times10^{-24}\ \approx\ 0$
Analytically this makes sense since no value of the function should never dip below 0.

2 Question 51

Minimum at $(u,\ v)\ =\ (0.0052m,\ 0.0284m)$
$V(u=0.0052m,\ v=0.0284m)\ \approx\ -0.10669300616237495J$

3 Question 52

Minimum at $(x,\ y,\ z)\ =\ (-0.1,\ 0.35,\ 0.05)$
$F(x=-0.1,\ y=0.35,\ z=0.05)\ \approx\ -0.35$
Analytically:

$$F(x,y,z)=2x^2+3y^2+z^2+xy+xz-2y \tag{3.1}$$

Minimum when,

$$\nabla F(x,y,z)=\mathbf{0} \tag{3.2}$$

$$\frac{\partial}{\partial x}F=4x+y+z=0 \tag{3.3}$$
$$\frac{\partial}{\partial y}F=6y+x-2=0 \tag{3.4}$$
$$\frac{\partial}{\partial z}F=2z+x=0 \tag{3.5}$$
$$\tag{3.6}$$

$$y=\frac{1}{3}-\frac{x}{6} \tag{3.7}$$
$$z=-\frac{x}{2} \tag{3.8}$$

$$4x+\left(\frac{1}{3}-\frac{x}{6}\right)+\left(-\frac{x}{2}\right)=0\rightarrow\boxed{x=-\frac{1}{10}=-0.1} \tag{3.9}$$
$$6y-\frac{1}{10}-2=0\rightarrow\boxed{y=\frac{7}{20}=0.35} \tag{3.10}$$
$$2z-\frac{1}{10}=0\rightarrow\boxed{z=\frac{1}{20}=0.05} \tag{3.11}$$

```
import numpy as np
from math import *
import matplotlib.pyplot as plt

def bracket(f,x1,h):
    c = (1+sqrt(5))/2#phi
    f1 = f(x1)
    x2 = x1+h
    f2 = f(x2)
    #Determine downhill directions and change sign(h) if needed
    if f2>f1:
        h = -h
        x2 = x1+h
        f2 = f(x2)#Change eval
        #Check if already at min
        if f2 > f1: return (x2,x1-h)
    #Search loop
    for i in range(100):
        h = c * h#update h
        x3 = x2+h#increment
        f3 = f(x3)
        if f3>f2: return (x1,x3)#check if passed
        x1=x2; x2 = x3; f1 = f2; f2 = f3
    print("Bracket did not find min")

def search(f, a, b, tol=1.0e-9):
  #determine num of iters
    nIter = int(ceil(-2.078087*log(tol/abs(b-a))))
    R = (1+sqrt(5))/2-1#phi
    C = 1.0 - R
    #First telescoping
    x1 = R*a + C*b; x2 = C*a + R*b
    f1 = f(x1); f2 = f(x2)
    for i in range(nIter):
        if f1 > f2:#search
            a = x1
            x1 = x2; f1 = f2
            x2 = C*a + R*b; f2 = f(x2)
        else:#search in other dir if needed
            b = x2
            x2 = x1; f2 = f1
            x1 = R*a + C*b; f1 = f(x1)
    if f1 < f2: return x1,f1
    else: return x2,f2
```

```
from hw1 import *

#def Q50():# Accidentally did page 400 p4
#    #P=VI, V=IR, P=I^2R
#    def f(R):
#        A = np.array( [[2 + 1.5 + R, -R],\
#                                    [-R, 3.6 + 1.8 + 1.2 + R]] )
#        B = np.array( [[120.0], [0.0]])
#        LU,P=LUdecomp(A)
#        X = LUsolve(A,B,LU,P)
#        P = R*(X[0]-X[1])**2
#        return -P
#
#    xStart = 0.1
#    h = 0.01
#    x1, x2 = bracket(f,xStart,h)
#    x, fMin = search(f,x1,x2)
#    print("R=",x)
#    print("P=",fMin)

def powell(F,x,h=0.1,tol=1.0e-6):
    def f(s): return F(x+s*v)#F in direction of v
    n = len(x)              #num of variables
    df = np.zeros(n)#Decreases of F stored
    u = np.identity(n)#Vectors v stored
    for j in range(30):
        xOld = x.copy()#for convergence
        fOld = F(xOld)
        for i in range(n):
            v = u[i]#update v
            a,b = bracket(f,0.0,h)#search for loc
            s,fMin = search(f,a,b)#search for loc2
            df[i] = fOld - fMin
            fOld = fMin
            x = x + s*v
        #Last search
        v = x - xOld
        a,b = bracket(f,0.0,h)
        s,fLast = search(f,a,b)
        x = x + s*v
        #check for convergence
        if sqrt(np.dot(x-xOld,x-xOld)/n) < tol: return x,j+1
        #identify biggest decrease and change search dir
        iMax = np.argmax(df)
```

```
        for i in range(iMax,n-1):
            u[i] = u[i+1]
        u[n-1]=v
    print("Powell did not converge")


#P50: Page 400 Problem 6
def P50():
    xStart = np.array([2.0,2.0])
    for lam  in [1.0, 10., 1000., 10000.]:#Dynamic penalty
        def F(x):
            #x+y>=1
            #x>=0.6
            #F=(x-1)^2+(y-1)^2
            #F(x)-constraints*penalty
            return (x[0]-1)**2+(x[1]-1)**2 + (max(1-x[0]-x[1],0)**2+max(0.6-
        x,numIter = powell(F,xStart,0.01)
        if np.sum(abs(x-xStart))<1.0e-6:#test for convergence
            break
        else:
            xStart=x
    print("Min=",x)
    print("F(x,y)=",F(x))

#P51: Page 402 Problem 13
#Write a function that minimizes V.
#Your function should return two variables, first the minimum value of
#V and second an array containing the values of u and v
#(in meters) that result in the minimum V.
def P51():
    xStart = np.array([0.,0.])
    for lam  in [1]:#[1.0, 10., 1000., 10000.]:
        def F(x):
            P,a,b,k,u,v=5,150/1000,50/1000,0.6*1000,x[0],x[1]#SI
            dab = sqrt((a+u)**2+v**2)-a
            dbc = sqrt((b-u)**2+v**2)-b
            return -P*v+k*(a+b)*(dab**2/a+dbc**2/b)/2
        x,numIter = powell(F,xStart,0.01)
        if np.sum(abs(x-xStart))<1.0e-9:
            break
        else:
            xStart=x
    print("Min=",x)
    print("F(u,v)=",F(x))
    return F(x), x
```

```
#P52: Page 403 Problem 17
#Write a function that returns the (x,y,z)
#found by minimizing the function. Also determine the (x,y,z)
#analytically (use the computed solution to find the analytical one).
def P52():
    xStart = np.array([0.,0.,0.])
    for lam  in [1]:#[1.0, 10., 1000., 10000.]:
        def F(x):
            x,y,z=x[0],x[1],x[2]
            return 2*x**2 + 3*y**2 + z**2 + x*y + x*z - 2*y
        x,numIter = powell(F,xStart,0.01)
        if np.sum(abs(x-xStart))<1.0e-9:
            break
        else:
            xStart=x
    print("Min=",x)
    print("F(x,y,z)=",F(x))



if __name__ == "__main__":
    P50()
    P51()
    P52()
```