MAE 598: design optimization

HOMEWORK – 4

Problem 1:

$$\min f(x) = (x_1 + 1)^2 + (x_2 - 2)^2$$

$$g_1 = x_1 - 2 \leq 0 \qquad g_2 = x_2 - 1 \leq 0$$

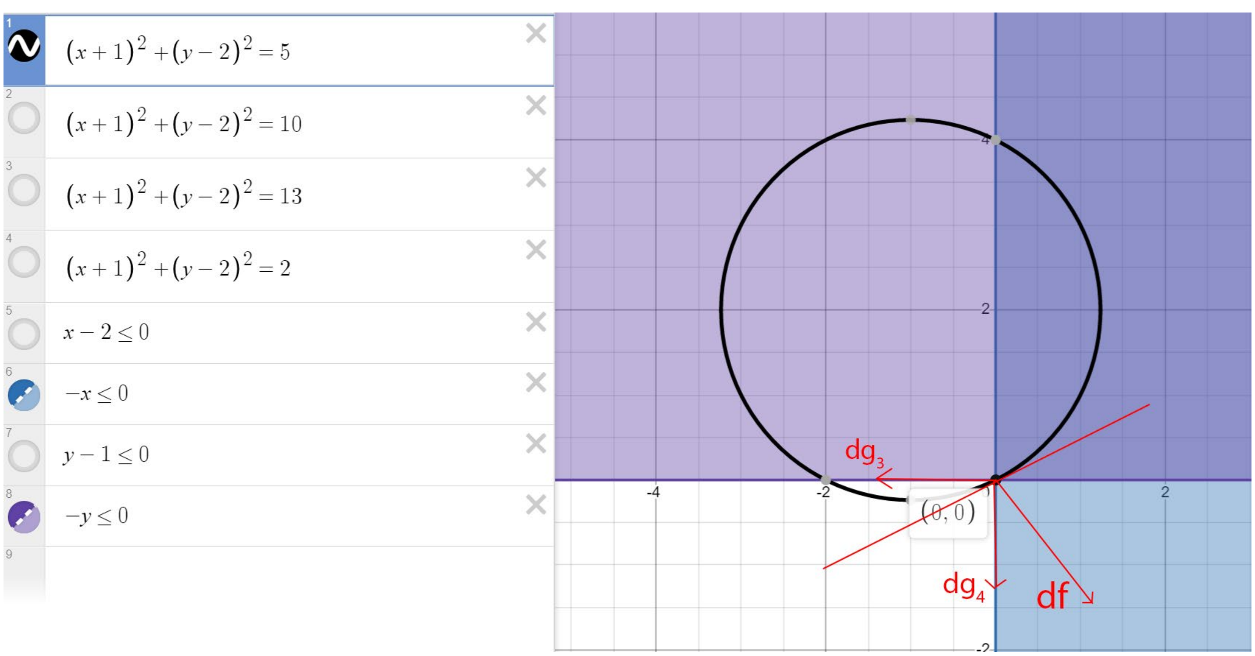
$$g_3 = -x_1 \leq 0 \qquad g_4 = -x_2 \leq 0$$

Taking $x_1 = 0$ and $x_2 = 0$

$$(x_1 + 1)^2 + (x_2 - 2)^2 = 5$$

For this $g_3 = -x_1 \leq 0$

$$g_4 = -x_2 \leq 0$$

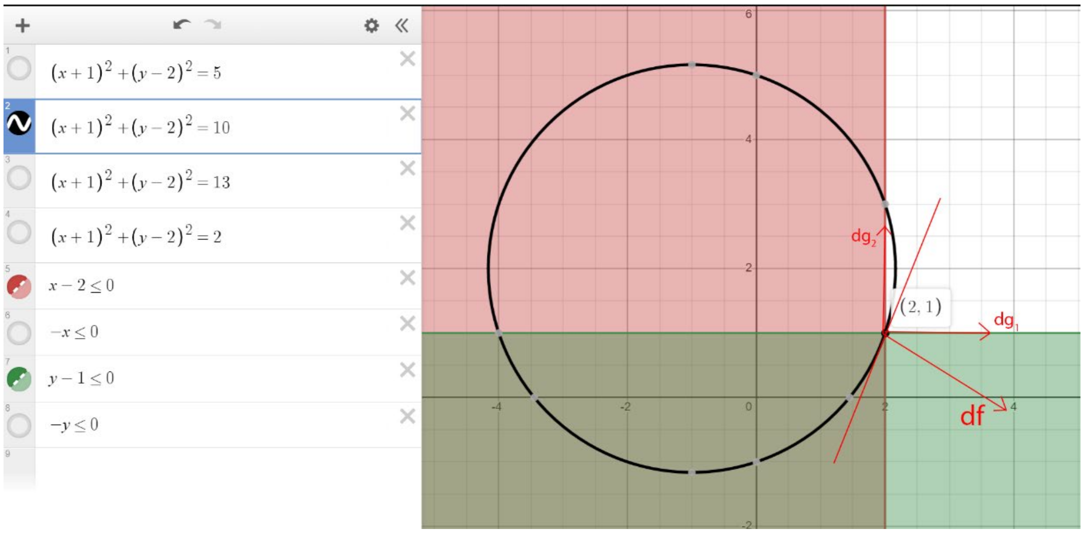


Taking $x_1 = 2$ and $x_2 = 1$

$$(x_1 + 1)^2 + (x_2 - 2)^2 = 10$$

For this:

$$g_1 = x_1 - 2 \leq 0 \qquad g_2 = x_2 - 1 \leq 0$$

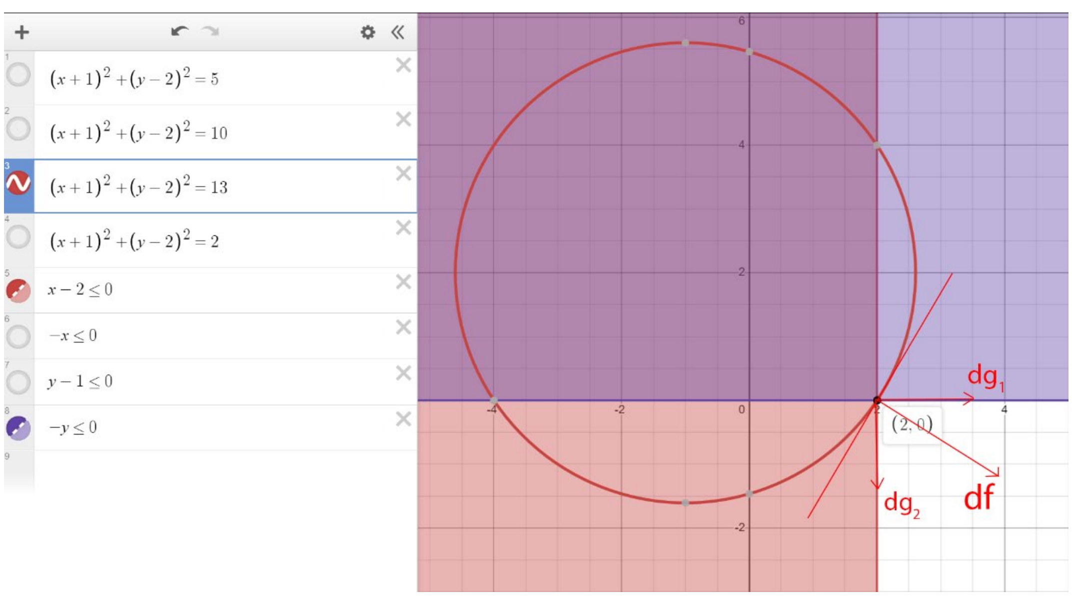


Taking $x_1 = 2$ and $x_2 = 0$

$$(x_1 + 1)^2 + (x_2 - 2)^2 = 13$$

For this:

$$g_1 = x_1 - 2 \leq 0 \quad g_4 = -x_2 \leq 0$$



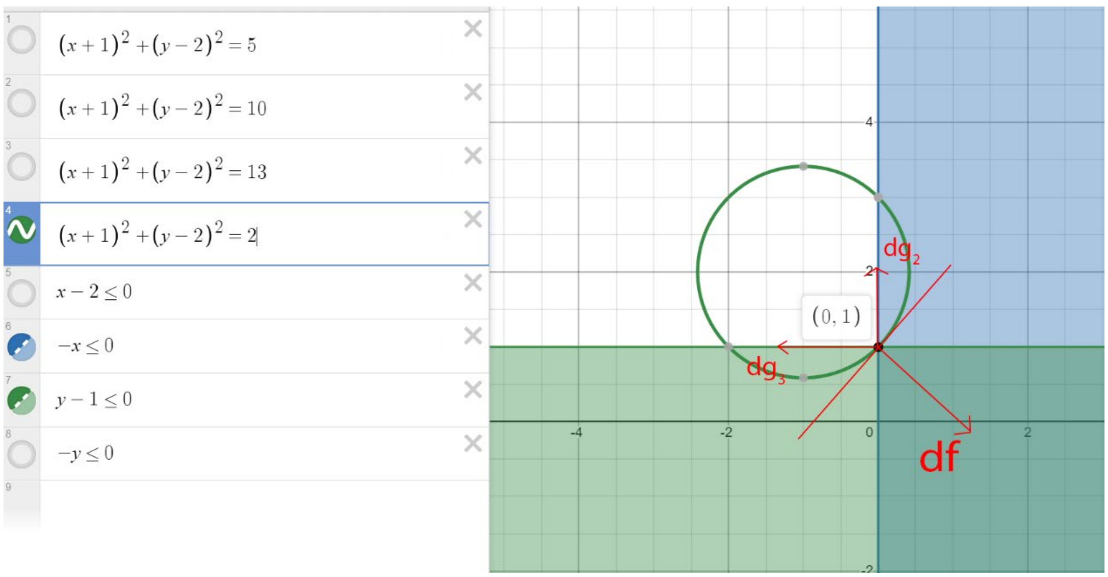
Taking $x_1 = 0$ and $x_2 = 1$

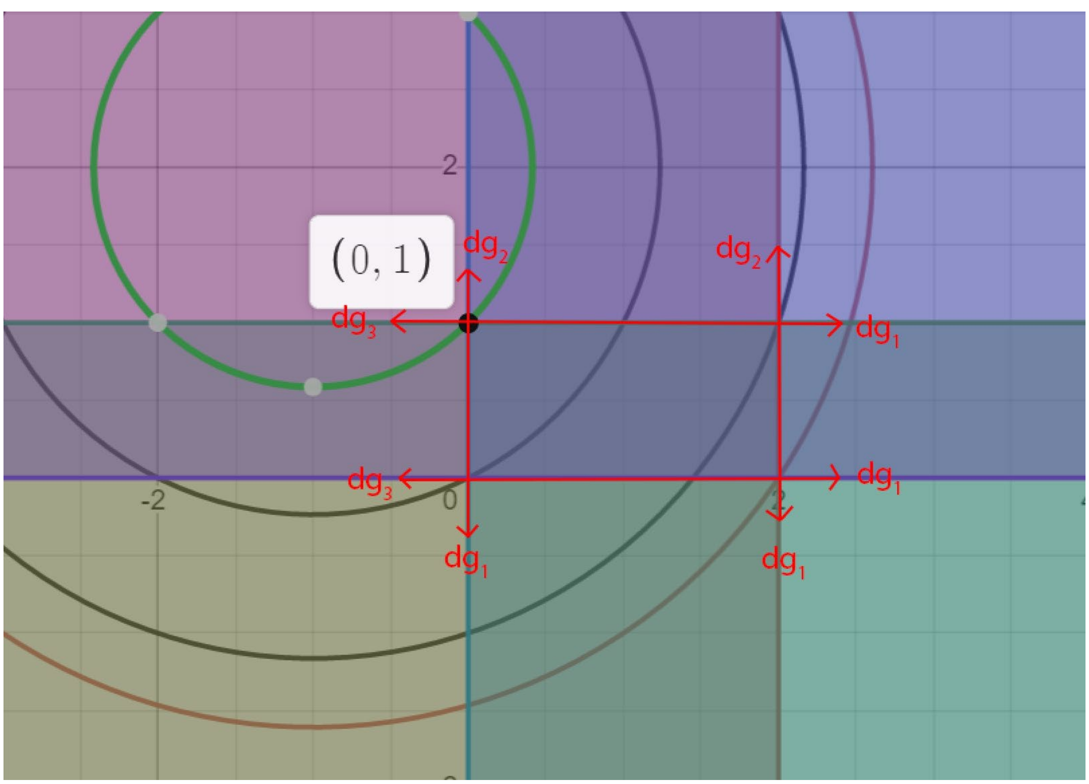
$$(\boldsymbol{x_1} + \mathbf{1})^2 + (\boldsymbol{x_2} - \mathbf{2})^2 = 2$$

For this:

$$\boldsymbol{g_2} = \boldsymbol{x_2} - \mathbf{1} \leq \mathbf{0}$$

$$\boldsymbol{g_3} = -\boldsymbol{x_1} \leq \mathbf{0}$$





Now, apply KKT conditions

$L = f + \lambda^T(equalities) + \hat{\mu}^T\{(inequalities)$

$L = (x_1 + 1)^2 + (x_2 - 2)^2 + \ \mu_1(x_1 - 2) + \ \mu_2(x_2 - 1) + \ \mu_3(-x_1) + \ \mu_4(-x_2)$

$$\nabla_x L = \begin{bmatrix} 2(x_1 + 1) + \ \mu_1 - \ \mu_3 \\ 2(x_2 - 2) + \ \mu_2 - \mu_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

For $X_1 = 0$ and $X_2 = 1$

SO, $\mu_3 > 0$, $\mu_2 > 0$, $\mu_1 = 0$, $\mu_4 = 0$

$$\nabla_x L = \begin{bmatrix} 2(1) - \ \mu_3 \\ 2(1 - 2) + \ \mu_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 2 - \ \mu_3 \\ -2 + \ \mu_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$\mu_2 = 2 > 0$

$\mu_3 = 2 > 0$

So, KKT conditions satisfied

$\nabla_{xx} L = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$, Hessian is positive definite

Hence, X =(0,1) is a global minimum

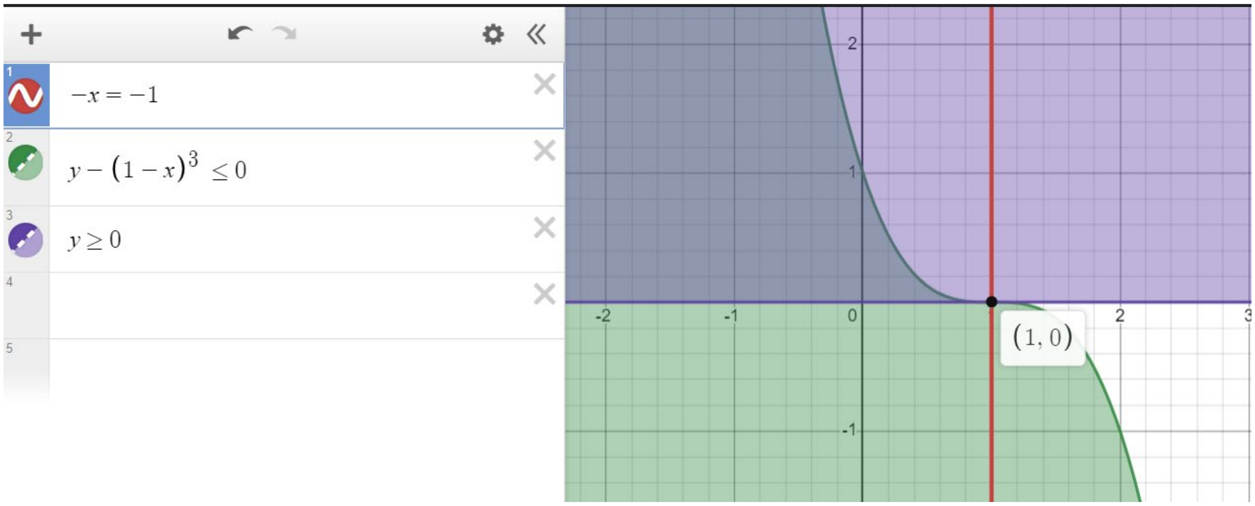
Problem 2

$\min f = -x_1$ Subjected to

$g_1 = x_2 - (1 - x_1)^3 \le 0$

$g_2 = x_2 \ge 0$

Find the solution graphically. Then apply the optimality conditions. Can you find a solution based on the optimality conditions? Why?



So, graphically we can observe that solution to this problem is X = (1,0)

Lagrangian:

$L = -x_1 + \mu_1(x_2 - (1 - x_1)^3) + \mu_2(-x_2)$

$$\nabla_x L = \begin{bmatrix} -1 + 3(1 - X_1)^2 \mu_1 \\ \mu_1 - \mu_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Now if we take

$X_1 = 1$ then $\mu_1 = 0$

$X_2 = 0$ then $\mu_2 > 0$

But from $\nabla_x L$ we know that $\mu_1 = \mu_2$,it cannot be true that $\mu_2 = 0$ & $\mu_2 > 0$

So, we cannot find solution using KKT conditions to this problem.

MAE 598: design optimization

HOMEWORK – 4

Problem 1:

$$\min f(x) = (x_1+1)^2 + (x_2-2)^2$$

$$g_1 = x_1 - 2 \le 0 \qquad g_2 = x_2 - 1 \le 0$$

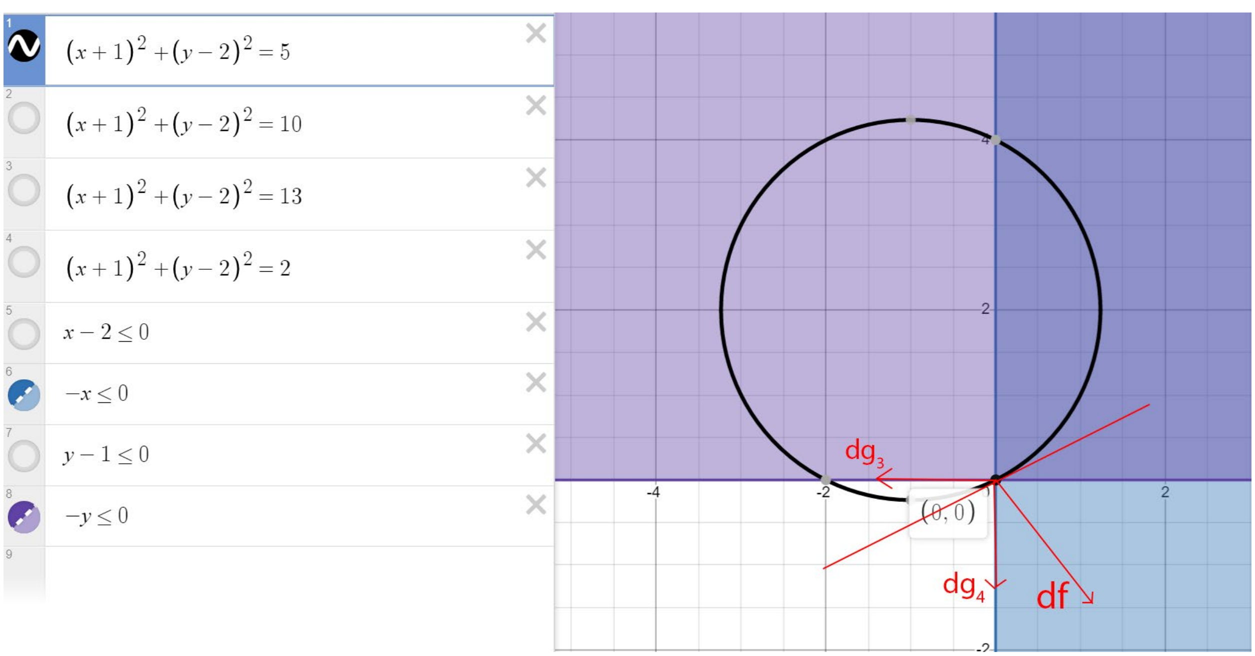
$$g_3 = -x_1 \le 0 \qquad g_4 = -x_2 \le 0$$

Taking $x_1 = 0$ and $x_2 = 0$

$$(x_1+1)^2 + (x_2-2)^2 = 5$$

For this $g_3 = -x_1 \le 0$

$$g_4 = -x_2 \le 0$$

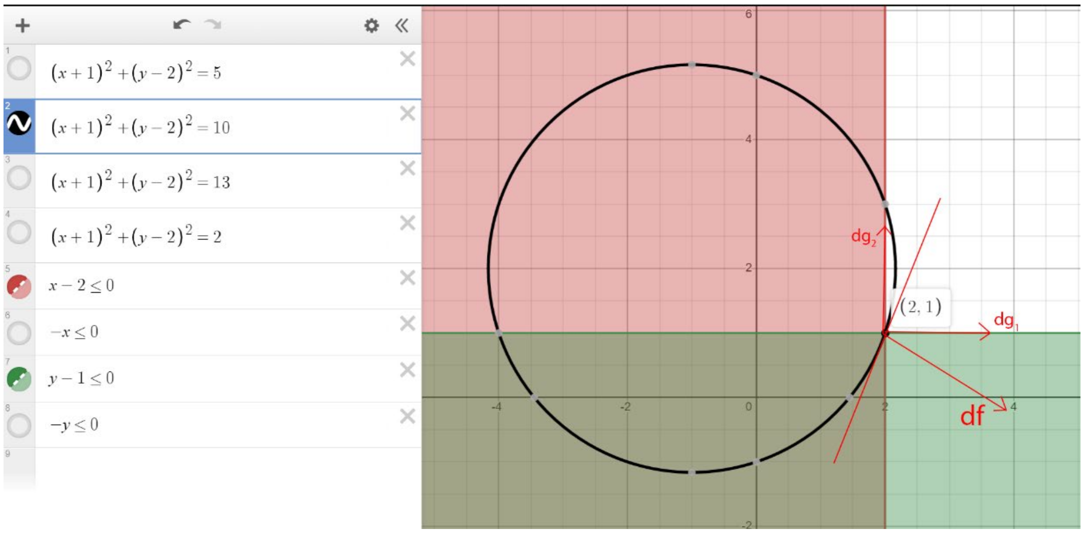


Taking $x_1 = 2$ and $x_2 = 1$

$$(x_1 + 1)^2 + (x_2 - 2)^2 = 10$$

For this:

$$g_1 = x_1 - 2 \leq 0 \qquad g_2 = x_2 - 1 \leq 0$$

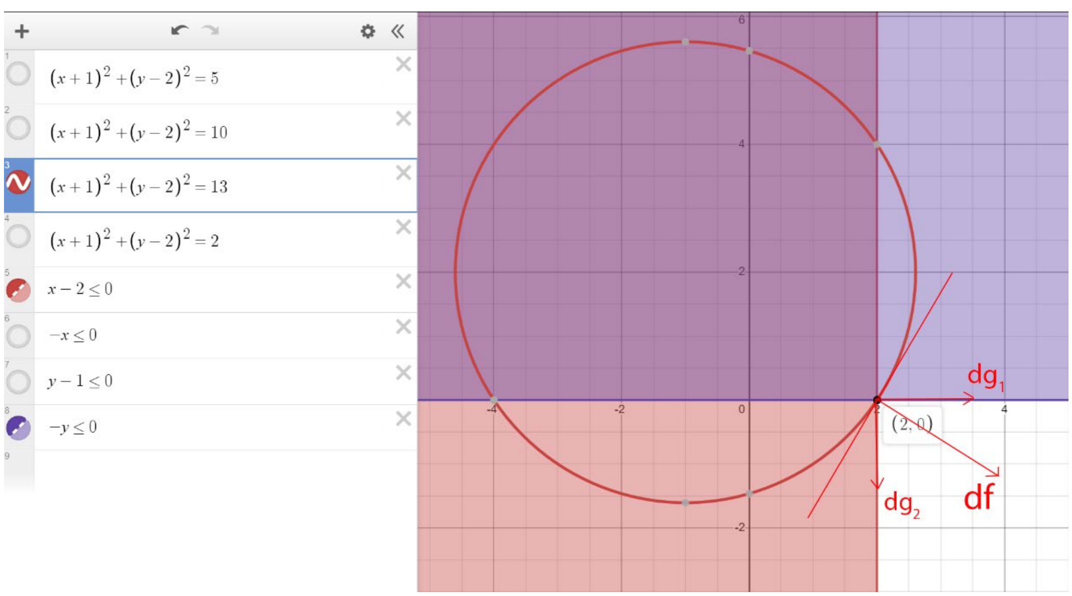


Taking $x_1 = 2$ and $x_2 = 0$

$$(x_1 + 1)^2 + (x_2 - 2)^2 = 13$$

For this:

$$g_1 = x_1 - 2 \leq 0 \quad g_4 = -x_2 \leq 0$$



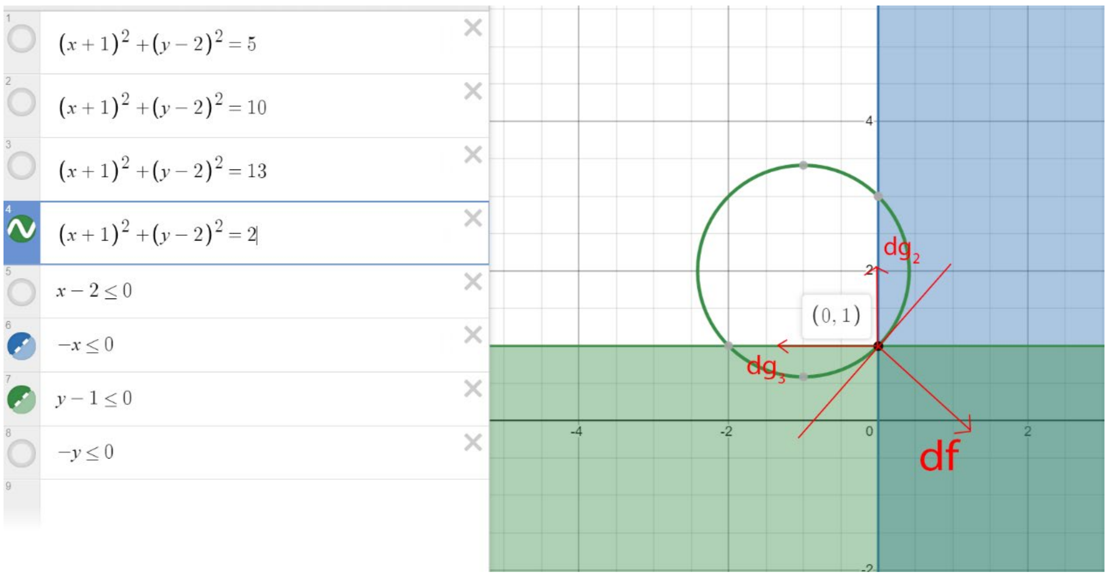
Taking $x_1 = 0$ and $x_2 = 1$

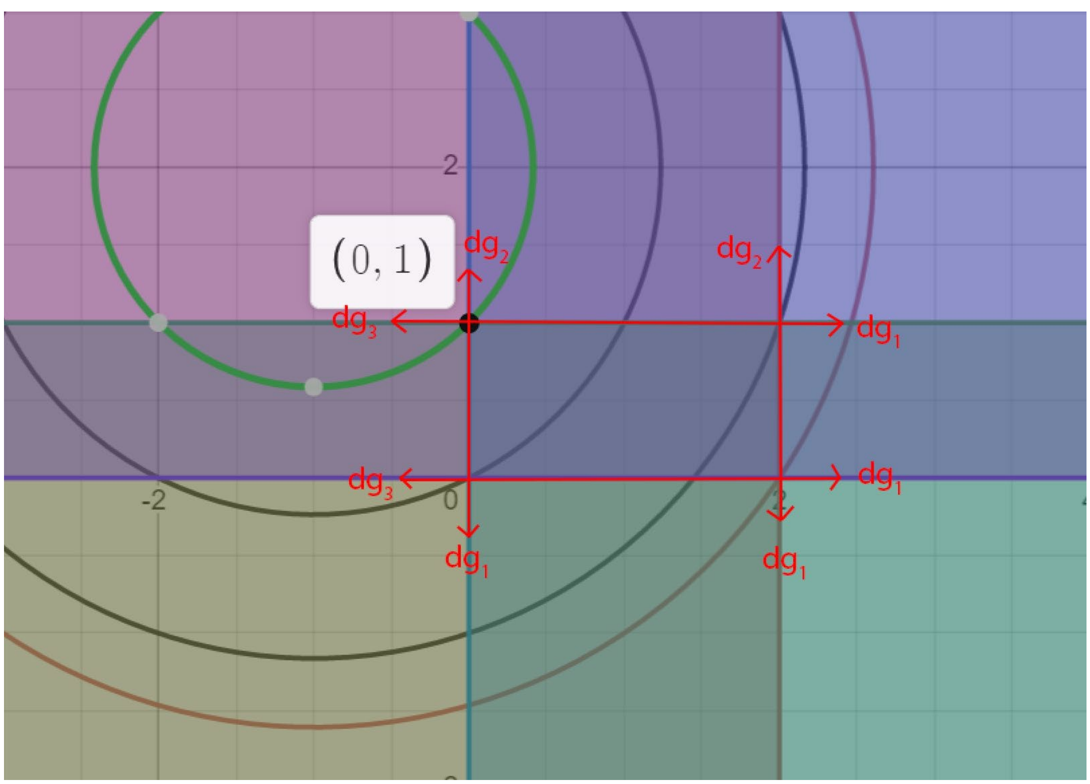
$$(\boldsymbol{x_1} + \mathbf{1})^2 + (\boldsymbol{x_2} - \mathbf{2})^2 = 2$$

For this:

$$\boldsymbol{g_2} = \boldsymbol{x_2} - \mathbf{1} \leq \mathbf{0}$$

$$\boldsymbol{g_3} = -\boldsymbol{x_1} \leq \mathbf{0}$$





Problem : 3

- find local solution to the problem

max $f = x_1x_2 + x_2x_3 + x_1x_3$

Subjected to $h = x_1 + x_2 + x_3 - 3 = 0$

⇒ Lagrangion method.

$L = (-x_1x_2) + (-x_2x_3) + (-x_1x_3) + \lambda(x_1 + x_2 + x_3 - 3)$

$L = -x_1x_2 - x_2x_3 - x_1x_3 + \lambda(x_1 + x_2 + x_3 - 3)$

$$\nabla_x L = \begin{bmatrix} -x_2 - x_3 + \lambda \\ -x_1 - x_3 + \lambda \\ -x_2 - x_1 + \lambda \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$-x_2 - x_3 + \lambda = -x_1 - x_3 + \lambda$

→ $x_2 = x_1$

$-x_1 - x_3 + \lambda = -x_2 - x_1 + \lambda$

→ $x_2 = x_3$

- hence $x_1 = x_2 = x_3$

$h = x_1 + x_2 + x_3 - 3 = 0$

$3x_1 = 3$

$\boxed{x_1 = x_2 = x_3 = 1}$

$\rightarrow \quad -x_2 - x_3 + \lambda = 0$

$$\lambda = x_2 + x_3$$

$$\boxed{\lambda = 2}$$

$\Rightarrow$ Reduced gradient method

$- \quad h = x_1 + x_2 + x_3 - 3 = 0$

$d = x_1, x_2$

$S = x_3 = 3 - x_1 - x_2$

$$= \frac{\partial f}{\partial d} - \left(\frac{\partial f}{\partial s}\right)\left(\frac{\partial h}{\partial s}\right)^{-1} \frac{\partial h}{\partial d}$$

$$\frac{\partial f}{\partial d} = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix} = \begin{bmatrix} -x_2 - x_3 \\ -x_1 - x_3 \end{bmatrix}$$

$$\frac{\partial f}{\partial s} = \begin{bmatrix} \frac{\partial f}{\partial x_3} \end{bmatrix} = \begin{bmatrix} -x_2 - x_1 \end{bmatrix}$$

$$\frac{\partial h}{\partial s} = \begin{bmatrix} \frac{\partial (x_1 + x_2 + x_3 - 3)}{\partial x_3} \end{bmatrix} = [1]$$

$$\frac{\partial h}{\partial b} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\frac{df}{dd} = \begin{bmatrix} -x_2 - x_3 \\ -x_1 - x_3 \end{bmatrix} - \begin{bmatrix} -x_2 & -x_1 \end{bmatrix} [1] \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} -x_2 - x_3 \\ -x_1 - x_3 \end{bmatrix} + \begin{bmatrix} x_2 + x_1 \\ x_2 + x_1 \end{bmatrix}$$

$$= \begin{bmatrix} x_1 - x_3 \\ x_2 - x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$x_1 = x_3$$
$$x_2 = x_3$$

hence, $\boxed{x_1 = x_2 = x_3}$

$$h = x_1 + x_2 + x_3 - 3 = 0$$

$$\boxed{x_1 = x_2 = x_3 = 1}$$

Problem-4

$$\min\ f = x_1^2 + x_2^2 + x_3^2$$

subjected to $h_1 = \frac{x_1^2}{4} + \frac{x_2^2}{5} + \frac{x_3^2}{25} - 1 = 0$

$h_2 = x_1 + x_2 - x_3 = 0$

- $n = 3$ decision variable $= x_1$
- $m = 2$ state variable $= x_2, x_3$
- $dof = n - m = 1$

$$\frac{df}{dd} = \frac{\partial f}{\partial d} + \frac{\partial f}{\partial s} \cdot \left(-\frac{\partial h}{\partial s}\right)^{-1} \cdot \frac{\partial h}{\partial d}$$

$$= \frac{\partial f}{\partial d} - \frac{\partial f}{\partial s}\left(\frac{\partial h}{\partial s}\right)^{-1} \cdot \frac{\partial h}{\partial d}$$

$$\frac{\partial f}{\partial d} = [2x_1]$$

$$\frac{\partial f}{\partial s} = [2x_2 \quad 2x_3]$$

$$\frac{\partial h}{\partial s} = \begin{bmatrix} \frac{2x_2}{5} & \frac{2x_3}{25} \\ 1 & -1 \end{bmatrix}$$

$$\frac{\partial h}{\partial d} = \begin{bmatrix} x_1/2 \\ 1 \end{bmatrix}$$

- Contiued in code...

```python
import numpy
import matplotlib.pyplot as plt
import math

#defining objective function , dfdd dfds dhds dhdd, DfDd
def fun(x):
    return  x[0] ** 2 + x[1] ** 2 + x[2] ** 2

def Dfdd(x):
    return 2*x[0]

def dfds(x):
    return numpy.array([2*x[1], 2*x[2]])

def dhds(x):
    return numpy.array([[2/5*x[1] , 2/25*x[2]], [1, -1]])

def dhdd(x):
    return numpy.array([[x[0]/2], [1]])

def DfDd(x):
    return Dfdd(x) - numpy.matmul(numpy.matmul(dfds(x),
numpy.linalg.inv(dhds(x))), dhdd(x))

def xe(x, a, dfdd):
    d_eval = (x[0] - a * dfdd)[0]
    s_eval = x[1:3] + a *
numpy.transpose(numpy.matmul(numpy.matmul(numpy.linalg.inv(dhds(x)),
dhdd(x)), numpy.transpose([DfDd(x)])))[0]
    return numpy.append(d_eval, s_eval)

#linesearch
def linesearch(dfdd, x):
    a = 0.5
    b = .8
    t = .5
    while fun(xe(x, a, dfdd)) > (fun(x) - a * t * dfdd ** 2):
        a = b * a
    return a

#intial value
x0 = numpy.array([0.5, 0.5, 0.25])

esp = 1e-03
xs = [x0]
error = []

def solution(x):
    #while numpy.linalg.norm(numpy.array(h_x)) > e:
```

```
    while numpy.linalg.norm(numpy.array([[x[0] ** 2 / 4 + x[1] ** 2 /
5 + x[2] ** 2 / 25 - 1], [x[0] + x[1] - x[2]]])) > esp:
        dh_ds = dhds(x)
        sk_1 = numpy.transpose(numpy.transpose([x[1:3]]) -
numpy.matmul(numpy.linalg.inv(dh_ds), numpy.array([[x[0] ** 2 / 4 +
x[1] ** 2 / 5 + x[2] ** 2 / 25 - 1],[x[0] + x[1] - x[2]]])))
        x = numpy.append(x[0:1], numpy.transpose(sk_1[0]))
    return x

while numpy.linalg.norm(DfDd(xs[-1])) > esp:
    x = xs[-1]
    df_dd = DfDd(x)
    error.append(math.log(numpy.linalg.norm(df_dd)))
    a = linesearch(df_dd, x)
    dk = x[0] - a * df_dd
    sk0 = x[1:3] + a *
numpy.transpose(numpy.matmul(numpy.matmul(numpy.linalg.inv(dhds(x)),
dhdd(x)), numpy.transpose(df_dd)))
    xk0 = numpy.append(dk, sk0)
    x = solution(xk0)
    xs.append(x)

print(xs[-1])
```

```
[-1.57420082  1.37758004 -0.19662078]
```

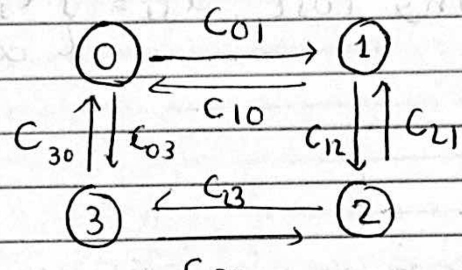
- Problem-5

- N sites to be visited and consider them as as nodes on graph

- cost of moving from node i to j is C_{ij}

- if there is an edge between nodes, or ∞ if there is none.

- Site 0 is the truck station where the truck starts and returns.



- Problem formulation

$$\min_{x_{ij}} \sum_{ij}^{N} x_{ij} c_{ij}$$

- forward $x_{ij} = \begin{cases} 1 & \text{if } i \text{ is connected with } j \\ 0 & \text{if } i \text{ is not connected with } j \end{cases}$

Similarly for

- backward motion $x_{ji} = \begin{cases} 1 & \text{if } j \text{ is connected with } i \\ 0 & \text{if } j \text{ is not connected with } i \end{cases}$

- forward moving cost $x_{ij} = \begin{cases} C_{ij} \\ \infty \end{cases}$

- backward moving cost $x_{ji} = \begin{cases} C_{ji} \\ \infty \end{cases}$

⇒ Constraints

$\sum x_{ij} > N$ truck need to visit all the nodes (locations)

$\sum x_{ij} = \sum_{ji}$

- There must be a connection with the neighboring node to start the process

$\sum x_{0j} \geq 1 \quad \forall j$ (for starting)

$\sum x_{j0} \geq 1 \quad \forall j$ (for ending)

$\Longrightarrow$ $\min_{x_{ij}} \sum_{ij}^{N} x_{ij} c_{ij}$

$\sum x_{ij} \geq N$

$\sum x_{ij} = \sum_{ji}$

$\sum x_{0j} \geq 1 \quad \forall j$

$\sum x_{j0} \geq 1 \quad \forall j$