Problem 1

(a) $g(y) = \|Cy + d\|_2^2$

$$= (Cy+d)^T(Cy+d)$$

$$= y^TC^TCy + 2y^TC^Td + d^Td$$

since $y \in \mathbb{R}^n$

$\Downarrow$

the dimension of the domain of g will be n

(b) let $C = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & & \vdots \\ c_{m1} & c_{m2} & \cdots & c_{mn} \end{bmatrix}$

$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$ $d = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_m \end{bmatrix}$

then $Cy+d = \begin{bmatrix} d_1 + \sum_{j=1}^{n} C_{1j} y_j \\ d_2 + \sum_{j=1}^{n} C_{2j} y_j \\ \vdots \\ d_m + \sum_{j=1}^{n} C_{mj} y_j \end{bmatrix}$

$\Downarrow$

$$g(y) = \| Cy + d \|_2^2$$

$$= \sum_{i=1}^{m} \left(d_i + \sum_{j=1}^{n} C_{ij} y_j \right)^2$$

$\Downarrow$

$$\partial g / \partial y_i = \sum_{k=1}^{m} 2 \left(d_k + \sum_{j=1}^{n} C_{kj} y_j \right) \cdot C_{ki}$$

$$= 2 \sum_{k=1}^{m} C_{ki} \left(d_k + \sum_{j=1}^{n} C_{kj} y_j \right)$$

for $i = 1, 2, \cdots, n$

$\partial^2 g / \partial y_i \partial y_j$

$$= \frac{\partial(\partial g/\partial y_i)}{\partial y_j} = \frac{\partial}{\partial y_j}\left[2\sum_{k=1}^{m} C_{ki}\left(d_k + \sum_{s=1}^{n} C_{ks} y_s\right)\right]$$

$$= 2\sum_{k=1}^{m} C_{ki} C_{kj}$$

for $j = 1, 2, \cdots, n$

(c) From (b), we can know that

$$\partial g/\partial y_i = 2\sum_{k=1}^{m} C_{ki}\left(d_k + \sum_{j=1}^{n} C_{kj} y_j\right)$$

$$= 2\sum_{k=1}^{m} C_{ki} d_k + 2\sum_{k=1}^{m} C_{ki} \sum_{j=1}^{n} C_{kj} y_j$$

then the gradient of g will be

$$\nabla g(y) = [\partial g/\partial y_1, \partial g/\partial y_2, \cdots, \partial g/\partial y_n]^T$$

$$= \begin{bmatrix} 2\sum_{k=1}^{m} C_{k1} d_k + 2\sum_{k=1}^{m} C_{k1} \sum_{j=1}^{n} C_{kj} y_j \\ 2\sum_{k=1}^{m} C_{k2} d_k + 2\sum_{k=1}^{m} C_{k2} \sum_{j=1}^{n} C_{kj} y_j \\ \vdots \\ 2\sum_{k=1}^{m} C_{kn} d_k + 2\sum_{k=1}^{m} C_{kn} \sum_{j=1}^{n} C_{kj} y_j \end{bmatrix}$$

$$= 2\begin{bmatrix} \sum_{k=1}^{m} C_{k1} d_k \\ \sum_{k=1}^{m} C_{k2} d_k \\ \vdots \\ \sum_{k=1}^{m} C_{kn} d_k \end{bmatrix} + 2\begin{bmatrix} \sum_{k=1}^{m} C_{k1} \sum_{j=1}^{n} C_{kj} y_j \\ \sum_{k=1}^{m} C_{k2} \sum_{j=1}^{n} C_{kj} y_j \\ \vdots \\ \sum_{k=1}^{m} C_{kn} \sum_{j=1}^{n} C_{kj} y_j \end{bmatrix}$$

$$= 2C^T d + 2C^T C y$$

$$= 2C^T (Cy + d)$$

From (b), we can also know that

$$\partial^2 g / \partial y_i \partial y_j$$

$$= 2\sum_{k=1}^{m} C_{ki} C_{kj}$$

then the Hessian of g will be

$$\nabla^2 g(y) = H = \begin{bmatrix} 2\sum_{k=1}^{m} C_{k1}C_{k1} & 2\sum_{k=1}^{m} C_{k1}C_{k2} & \cdots & 2\sum_{k=1}^{m} C_{k1}C_{kn} \\ 2\sum_{k=1}^{m} C_{k2}C_{k1} & 2\sum_{k=1}^{m} C_{k2}C_{k2} & \cdots & 2\sum_{k=1}^{m} C_{k2}C_{kn} \\ \vdots & \vdots & & \vdots \\ 2\sum_{k=1}^{m} C_{kn}C_{k1} & 2\sum_{k=1}^{m} C_{kn}C_{k2} & \cdots & 2\sum_{k=1}^{m} C_{kn}C_{kn} \end{bmatrix}$$

$= 2C^T C$

(d) From (a) we know the quadratic form of $g(y)$:

$$g(y) = y^T C^T C y + 2y^T C^T d + d^T d$$

then the quadratic form at $y+h$ will be:

$$g(y+h) = (y+h)^T C^T C(y+h) + 2(y+h)^T C^T d + d^T d$$

$$= (y^T C^T C y + 2y^T C^T d + d^T d) + (2y^T C^T C + 2d^T C)h + h^T C^T C h$$

$$= g(y) + \nabla g(y)^T h + \frac{1}{2} h^T \nabla^2 g(y) h + O(\|h\|_2^3)$$

$$= D(y) + E(y)h + h^T F(y) h + O(\|h\|_2^3)$$

As a result, we get

$$\begin{cases} g(y) = D(y) \\ \nabla g(y)^T h = (2y^T C^T C + 2d^T C) h = E(y) h \\ \frac{1}{2} h^T \nabla^2 g(y) h = h^T C^T C h = h^T F(y) h \\ O(\|h\|_2^3) = 0 \end{cases}$$

$$\Downarrow$$

$$\begin{cases} D(y) = y^T C^T C y + 2y^T C^T d + d^T d \\ E(y) = 2y^T C^T C + 2d^T C \\ F(y) = C^T C \end{cases}$$

$$\Downarrow$$

$$\begin{cases} \nabla g(y) = E(y)^T = 2C^T(Cy + d) \\ \nabla^2 g(y) = 2F(y) = 2C^T C \end{cases}$$

Problem 2

(a) $f(x,y) = (1-\cos(\pi x/2))y^2$

$(x,y) \in B = [-1,1] \times [-1,1]$

$\Downarrow$

$$\begin{cases} \partial f/\partial y = 2(1-\cos(\pi x/2))y \\ \partial f/\partial x = [\sin(\pi x/2)] \cdot \frac{\pi}{2} y^2 \end{cases}$$

let $\nabla f = (\partial f/\partial x, \partial f/\partial y) = (0,0)$

then $\begin{cases} 2(1-\cos(\pi x/2))y = 0 \\ [\sin(\pi x/2)] \cdot \frac{\pi}{2} y^2 = 0 \end{cases}$

$\Rightarrow$ this will be true for all $x=0$ in B and for all $y=0$ in B

so its set of minimizers over B will be

$$S = [-1,1] \times \{0\} \cup \{0\} \times [-1,1]$$

and the value of f will be

$$f(x,y) = 0, \quad \forall (x,y) \in S$$

HW1 Problem2 for Optimization

September 18, 2022

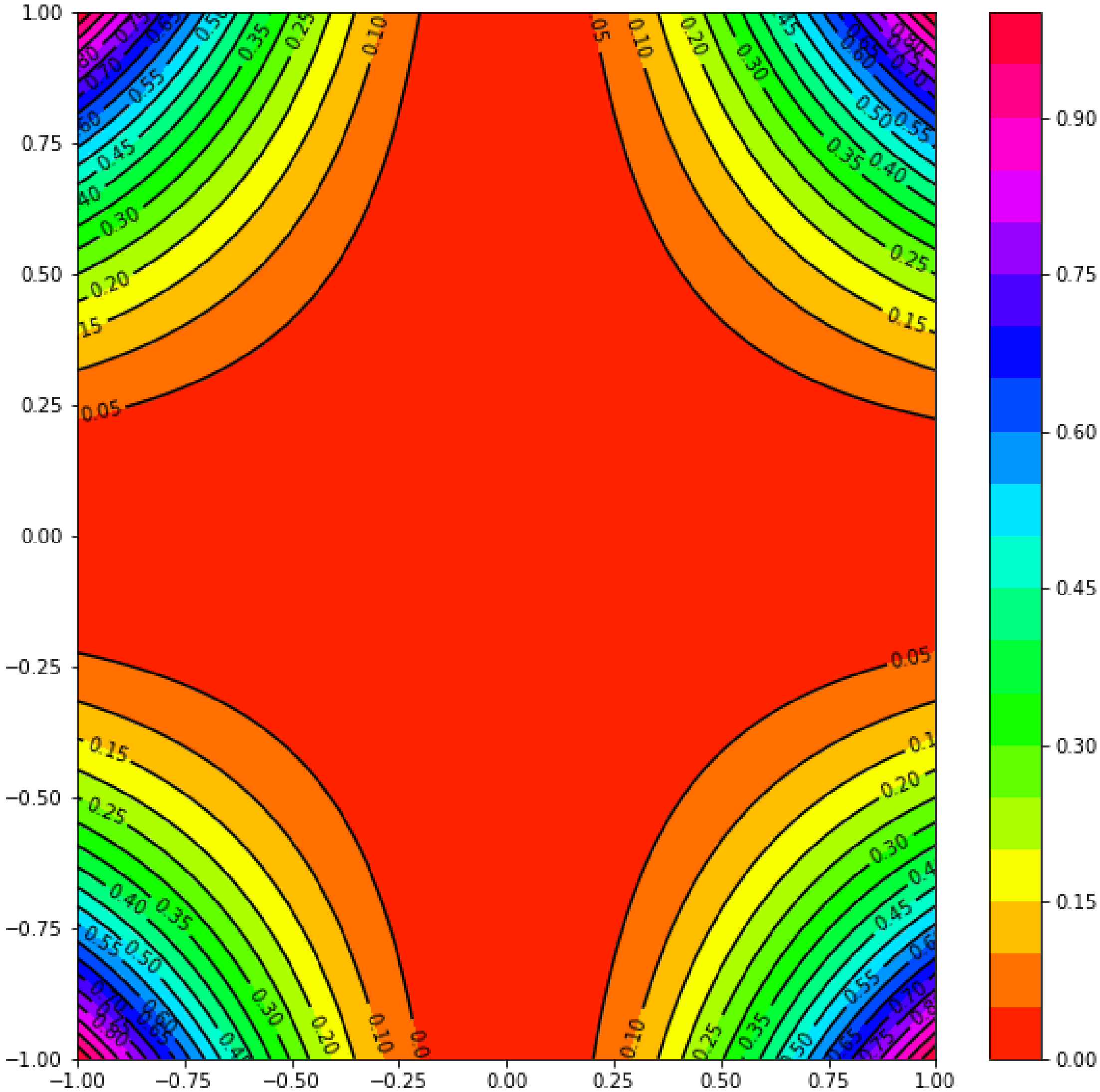
1 Prob2 (b)

[1]:
```
from matplotlib import pyplot as plt
import numpy as np

plt.figure(figsize = (10, 10))
x = np.linspace(-1, 1, 1000)
y = np.linspace(-1, 1, 1000)
x, y = np.meshgrid(x, y)
z = (1 - np.cos(np.pi * x / 2)) * y ** 2

ctf = plt.contourf(x, y, z, 20, cmap='hsv')
ct = plt.contour(x, y, z, 20, colors='k')
plt.clabel(ct, inline=True, fontsize=10)
plt.colorbar(ctf)
```

[1]:
```
<matplotlib.colorbar.Colorbar at 0x7f879a312490>
```



2 Prob2 (c)

[2]:
```
from matplotlib import pyplot as plt
import numpy as np

plt.figure(figsize = (10, 10))
x = np.linspace(-1, 1, 1000)
y = np.linspace(-1, 1, 1000)
x, y = np.meshgrid(x, y)
z = (1 - np.cos(np.pi * x / 2)) * y ** 2

r = np.random.uniform(-1, 1, 2)
```

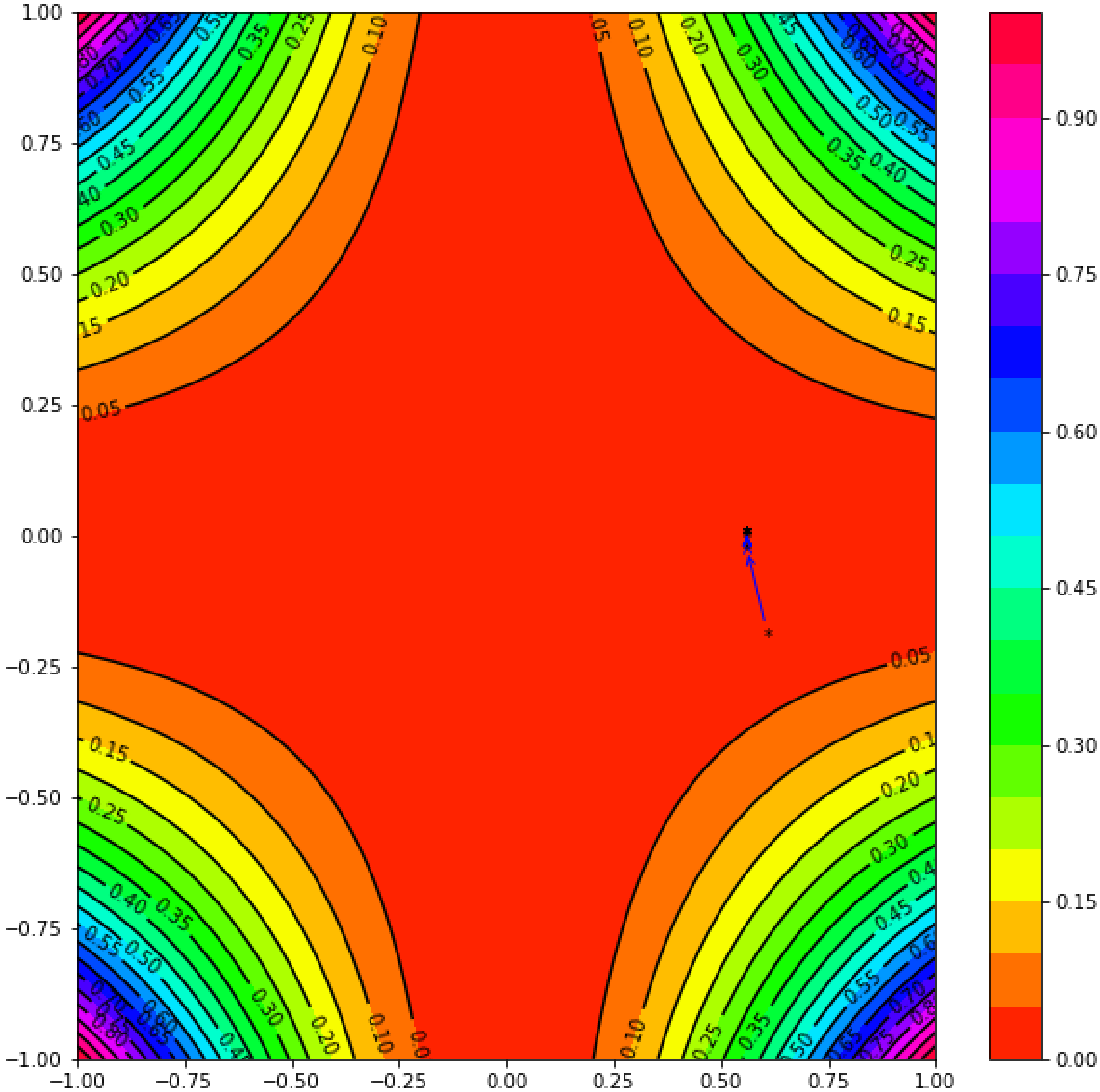
```python
k = 10

def grad_f(x, y):
    r1 = (np.pi / 2) * (y ** 2) * np.sin(np.pi * x / 2)
    r2 = 2 * y * (1 - np.cos(np.pi * x / 2))
    return x - r1, y - r2

for i in range(k):
    q = r
    r = grad_f(r[0], r[1])
    plt.annotate('*', xy = r, xytext = q,
             arrowprops = {'arrowstyle': '->', 'color': 'b', 'lw': 1},
             va = 'center', ha = 'center')

ctf = plt.contourf(x, y, z, 20, cmap='hsv')
ct = plt.contour(x, y, z, 20, colors='k')
plt.clabel(ct, inline=True, fontsize=10)
plt.colorbar(ctf)
```

```
[2]: <matplotlib.colorbar.Colorbar at 0x7f879a367160>
```



3 Prob2 (e)

[6]:
```
from matplotlib import pyplot as plt
import numpy as np

plt.figure(figsize = (10, 10))
x = np.linspace(-1, 1, 1000)
y = np.linspace(-1, 1, 1000)
x, y = np.meshgrid(x, y)
z = (1 - np.cos(np.pi * x / 2)) * y ** 2

r = np.random.uniform(-1, 1, 2)
```

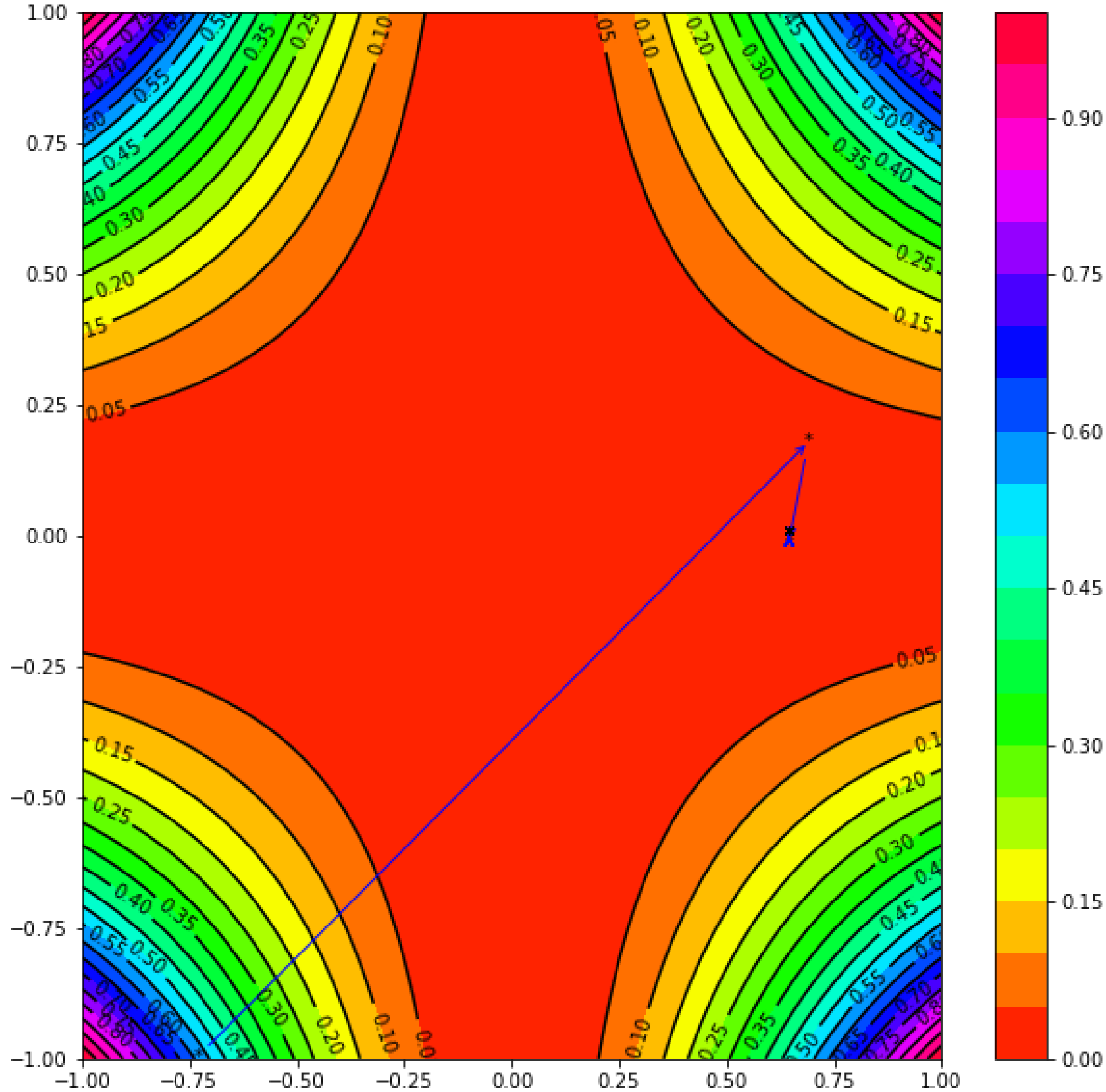
```python
k = 1000

def grad_f(x, y):
    r1 = (np.pi / 2) * (y ** 2) * np.sin(np.pi * x / 2)
    r2 = 2 * y * (1 - np.cos(np.pi * x / 2))
    return x - r1, y - r2

for i in range(k):
    q = r
    r = grad_f(r[0], r[1])
    plt.annotate('*', xy = r, xytext = q,
             arrowprops = {'arrowstyle': '->', 'color': 'b', 'lw': 1},
             va = 'center', ha = 'center')

ctf = plt.contourf(x, y, z, 20, cmap='hsv')
ct = plt.contour(x, y, z, 20, colors='k')
plt.clabel(ct, inline=True, fontsize=10)
plt.colorbar(ctf)
```

[6]: <matplotlib.colorbar.Colorbar at 0x7f879bce41f0>


1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
0.90
0.75
0.60
0.45
0.30
0.15
0.00