Introduction

The Primal-Dual Interior Point Method is an efficient way to solve constrained optimization problem by following the trajectory of interior point that goes through the interior of feasible set. For each iteration, Newton-Raphson method is used to solve the KKT system. The Predictor-Corrector Variant of the Primal-Dual Interior Point Method is an improvement on practical performance which finds a predictor step without centering and a corrector step that uses the second order approximation of the central path. To solve a constrained quadratic optimization problem in the form of (1), the coefficient matrix needs to be constructed as matrix (2) where Z is the diagonal matrix of z and X is the diagonal matrix of x. For each iteration, the affine scaling direction, d^{aff}, will be calculated by solving the system of equations (3) and the corresponding step length will be determined by minimum ratio tests (4). The centering parameter, τ, will be calculated using formula (5), where y_{aff} is the duality measure. Then another system of equation (6) needs to be solved to get the corrector direction and the step length (7) is determined by the same minimum ratio test (4) and damping factor η. Finally the next iteration can be obtained using formula (8). Due to the limited precision of computer, stopping criterions (9) needs to be implemented to stop the program when the current solution is close enough to the optimal solution where ε is the tolerance. The Primal-Dual Interior Point Method was implemented in Python and a constrained quadratic optimization problem was used to test the implemented function. The code for the implemented function and the test function is shown in Appendix A.

$$minimize\ c^T x + \frac{1}{2} x^T Q x$$

$$subject\ to\ Ax = b, x \geq 0 \qquad (1)$$

$$\begin{bmatrix} -Q & A^T & I \\ A & 0 & 0 \\ Z & 0 & X \end{bmatrix} \qquad (2)$$

$$\begin{bmatrix} -Q & A^T & I \\ A & 0 & 0 \\ Z^k & 0 & X^k \end{bmatrix} \begin{bmatrix} d_x^{aff} \\ d_\pi^{aff} \\ d_z^{aff} \end{bmatrix} = \begin{bmatrix} -r_d^{(k)} \\ -r_p^{(k)} \\ -X^{(k)} Z^{(k)} e \end{bmatrix} \qquad (3)$$

where $r_d^{(k)} = -Q^T x^{(k)} + A^T \pi^{(k)} + z^{(k)} - c,\ r_p^{(k)} = Ax^{(k)} - b$

$$\alpha^{aff} = min\left\{1, \min_{i:(d_x^{aff})_i<0} -\frac{x_i}{\left(d_x^{aff}\right)_i}, \min_{i:(d_z^{aff})_i<0} -\frac{z_i}{\left(d_z^{aff}\right)_i}\right\} \qquad (4)$$

$$\tau = \left(\frac{y_{aff}}{y}\right)^3 \qquad (5)$$

where $y_{aff} = \frac{\left(x+\alpha_x^{aff} d_x^{aff}\right)^T \left(z+\alpha_z^{aff} d_z^{aff}\right)}{n},\ y = \frac{x^T z}{n}$

$$\begin{bmatrix} -Q & A^T & I \\ A & 0 & 0 \\ Z^k & 0 & X^k \end{bmatrix} \begin{bmatrix} d_x \\ d_\pi \\ d_z \end{bmatrix} = \begin{bmatrix} -r_d^{(k)} \\ -r_p^{(k)} \\ -X^{(k)}Z^{(k)}e + D_x D_z e + \tau y e \end{bmatrix} \quad (6)$$

$$\alpha = min\{1, \eta\alpha_x^{max}, \eta\alpha_z^{max}\} \quad (7)$$

$$\begin{aligned} x^{(k+1)} &= x^k + \alpha d_x \\ \pi^{(k+1)} &= \pi^k + \alpha d_\pi \\ z^{(k+1)} &= z^k + \alpha d_z \end{aligned} \quad (8)$$

$$\begin{aligned} \|Ax - b\| &< \varepsilon \\ \|-Qx + A^T\pi + z - c\| &< \varepsilon \\ x^T z &< \varepsilon \end{aligned} \quad (9)$$

Result

The constrained quadratic optimization problem that was used to test the implemented function is defined as below:

$$c = [0,0,0]^T, Q = \begin{bmatrix} 0.02778 & 0.00387 & 0.00021 \\ 0.00387 & 0.01112 & -0.0002 \\ 0.00021 & -0.0002 & 0.00115 \end{bmatrix},$$

$$A = \begin{bmatrix} 0.1073 & 0.0737 & 0.0627 \\ 1 & 1 & 1 \end{bmatrix}, b = \begin{bmatrix} 0.0650 \\ 1 \end{bmatrix}$$

The initial infeasible solutions are:

$$x^{(0)} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \pi^{(0)} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, z^{(0)} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

The tolerance was set to be 10^{-8} and the damping parameter used to test the function is 0.95. The results of each iteration are shown in Table 1, Table 2 and Table 3, the output of each iteration is shown in Appendix B. From the result, it shows that the gap between primal function and dual function is iteratively decreasing and the gap is very small when the algorithm converged to the solution that satisfies the stopping criteria. With the founded solution, the value of the dual function is smaller than the value of the primal function, which proves that the dual function yields lower bounds on the optimal value of the primal problem.

Table 1: Primal Solution

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$c^T x + \frac{1}{2}x^T Q x$
0	1	1	1	0.023905
1	0.05	0.551804	0.62333086	0.00199561542
2	0.0025	0.50919873	0.61635703	0.0016026228
3	1.25000000e-04	3.44997068e-01	7.11183476e-01	0.00090370690
4	5.55787214e-04	2.34345151e-01	7.76453087e-01	0.00061620514
5	5.24183749e-04	2.08462037e-01	7.91631453e-01	0.00056946772

6	5.20649138e-04	2.07057629e-01	7.92453800e-01	0.00056715424
7	6.00284476e-04	2.06674214e-01	7.92732595e-01	0.0005666530
8	0.03386353	0.07178969	0.89434678	0.00050742759
9	0.03342589	0.07356413	0.89300998	0.00050679737
10	0.03079954	0.08421276	0.8849877	0.00050380465
11	0.02733624	0.09825487	0.87440888	0.00050192686
12	0.02646032	0.10180632	0.87173335	0.00050182474
13	0.02631524	0.10239458	0.87129018	0.00050182235
14	0.02630455	0.10243791	0.87125754	0.00050182234

Table 2: Dual Solution

k	$\pi_1^{(k)}$	$\pi_2^{(k)}$	$-b^T\pi - \frac{1}{2}x^TQx$
0	1	1	1.04109499
1	-6.81522183	-0.04482545	-0.489810482
2	-20.16128226	1.02286603	-0.28921993
3	-45.90249572	2.90202283	-0.082543102
4	-35.13799665	2.21395942	-0.070626570
5	-9.17933441	0.57696261	-0.0202635902
6	-0.4928369	0.0317996	-0.00080195119
7	-0.02216286	0.00226355	0.00025631218
8	-0.13819711	0.00961943	0.000129189316
9	-0.02757912	0.00273556	0.000436121231
10	-0.01150857	0.00172353	0.000471672
11	0.00290613	0.00080916	0.00049612893
12	0.00658834	0.00057457	0.00050098957
13	0.00719818	0.0005357	0.00050176272
14	0.00724311	0.00053284	0.00050181926

Table 3: Residuals and τ

k	τ	$\|Ax - b\|$	$\|-Q^Tx + A^T\pi + z - c\|$	x^Tz
0	-	2.00796755	3.57722081	3
1	0.0058940699	0.226031748	0.40267855	0.92321394
2	0.0462999445	0.128565908	0.22904187	0.53933504
3	0.016294248	0.0565298528	0.10070868	0.21225815
4	0.017534098	0.0113992568	0.0203079272	0.09358510
5	0.00160460368	0.00062013415	0.00110477721	0.021620188
6	1.70830482e-06	3.22053195e-05	5.73742036e-05	0.00140255491
7	0.0061930379	7.12190378e-06	1.26877660e-05	0.000317648939
8	0.50389952	0.0	3.67073e-18	0.00037823828
9	0.00131753259	0.0	2.60907267e-18	7.0676147e-05
10	0.0211553611	0.0	8.4814394e-19	3.2132653e-05
11	0.0113442296	0.0	6.5903790e-19	5.797928e-06

12	0.000347541639	0.0	2.2143689e-19	8.3516887e-07
13	1.62546829e-06	0.0	3.2044914e-19	5.9625179e-08
14	6.4289912e-10	0.0	4.8241683e-19	3.07777478e-09

Appendix A

Implemented function:

```
def primal_dual_interior_point(c,Q,A,b,x,pi,z,eta):
    """
    Description:
    this function uses primal-dual interior method to
    solve convex quadratic constrained optimization
    Input:
    c: coefficient of linear part
    Q: coefficient of quadratic part
    A: coefficient of linear constraint
    b: value of linear constraint
    x, pi, z: column vectors of initial infeasible solution
    eta: damping parameter
    Output:
    Optimal solution of x,pi,z
    """
    # import library
    import numpy as np
    # set tolerance
    tol=1E-8
    # number of variables
    var_num=len(x)
    # calcualte stopping criteria
    c_1=np.linalg.norm(np.dot(A,x)-b)
    c_2=np.linalg.norm(-np.dot(Q,x)+np.dot(A.T,pi)+z-c)
    c_3=np.dot(x.T,z)
    # initialize counter
    counter=0
    # print the information of initial infeasible solution
    print("=====Iteration{}=====".format(counter))
    print('x: {}'.format(x))
    print('pi: {}'.format(pi))
    print('z: {}'.format(z))
    print('tau: {}'.format('-'))
    print('primal problem: {}'.format(np.dot(c.T,x)+0.5*np.dot(x.
T,Q).dot(x)))
    print('dual problem: {}'.format(np.dot(b.T,pi)-0.5*np.dot(x.T
,Q).dot(x)))
    print('Residual of primal: {}'.format(c_1))
    print('Residual of dual: {}'.format(c_2))
    print('x.T dot x: {}'.format(c_3))
    # check stopping criteria
    while (c_1 > tol) | (c_2>tol) | (c_3>tol):
```

```
        # calculate residules for the infeasible solution
        r_p=np.dot(A,x)-b
        r_d=-np.dot(Q,x)+np.dot(A.T,pi)+z-c
        XZe=np.dot(np.diag(x),np.diag(z).dot(np.ones((var_num))))
        # stack the coefficient matrix
        coef_1=np.vstack((-Q,A,np.diag(z)))
        coef_2=np.vstack((A.T,np.zeros((coef_1.shape[0]-var_num,A
.shape[0]))))
        coef_3=np.vstack((np.identity(var_num),np.zeros((A.shape[
0],var_num)),np.diag(x)))
        coef=np.hstack((coef_1,coef_2,coef_3))
        # stack the residuals
        r=np.vstack((-r_d.reshape(-1,1),-r_p.reshape(-1,1),-XZe.r
eshape(-1,1)))
        # solve for affine scaling directions
        d_aff=np.linalg.solve(coef,r)
        # get direction for each part
        d_x_aff=d_aff[:var_num]
        d_pi_aff=d_aff[var_num:coef.shape[0]-var_num]
        d_z_aff=d_aff[-var_num:]
        # find step length of the affine scaling direction
        # create a list to store possible selections
        selections=[1]
        # iterate through possible selections and append result t
o the list
        for i in range(var_num):
            if d_x_aff[i] < 0:
                selections.append((-x[i]/d_x_aff[i])[0])
            if d_z_aff[i] < 0:
                selections.append((-z[i]/d_z_aff[i])[0])
        # find the minimum value in possible selections
        selections=np.array(selections)
        alpha_aff=np.min(selections)
        # calculate the duality measure
        y=np.dot(x.T,z)/var_num
        y_aff=np.dot((x+alpha_aff*d_x_aff.reshape(-1)).T,(z+alpha
_aff*d_z_aff.reshape(-1)))/var_num
        # calculate the centering parameter
        tau=(y_aff/y)**3
        # stack the adjusted residuals
        r_l=-XZe.reshape(-1,1)+np.diag(d_x_aff.reshape(-1)).dot(n
p.diag(d_z_aff.reshape(-1))).dot(np.ones((var_num,1)))+tau*y*np.o
nes((var_num,1))
```

```
        r_adjust=np.vstack((-r_d.reshape(-1,1),-r_p.reshape(-1,1)
,r_l))
        # solve for search direction
        d=np.linalg.solve(coef,r_adjust)
        # get direction for each part
        d_x=d[:var_num]
        d_pi=d[var_num:coef.shape[0]-var_num]
        d_z=d[-var_num:]
        # create a list to store possible selections
        selections=[1]
        # iterate through possible selections and append result t
o the list
        for i in range(var_num):
            if d_x[i] < 0:
                selections.append((-eta*x[i]/d_x[i])[0])
            if d_z[i] < 0:
                selections.append((-eta*z[i]/d_z[i])[0])
        # find the minimum value in possible selections
        selections=np.array(selections)
        alpha=np.min(selections)
        # calcualte new iterates
        x=x+alpha*d_x.reshape(-1)
        pi=pi+alpha*d_pi.reshape(-1)
        z=z+alpha*d_z.reshape(-1)
        # calcualte stopping criteria
        c_1=np.linalg.norm(np.dot(A,x)-b)
        c_2=np.linalg.norm(-np.dot(Q,x)+np.dot(A.T,pi)+z-c)
        c_3=np.dot(x.T,z)
        # print the information of each iteration
        print("=====Iteration{}=====".format(counter+1))
        print('x: {}'.format(x))
        print('pi: {}'.format(pi))
        print('z: {}'.format(z))
        print('tau: {}'.format(tau))
        print('primal problem: {}'.format(np.dot(c.T,x)+0.5*np.do
t(x.T,Q).dot(x)))
        print('dual problem: {}'.format(np.dot(b.T,pi)-0.5*np.dot
(x.T,Q).dot(x)))
        print('Residual of primal: {}'.format(c_1))
        print('Residual of dual: {}'.format(c_2))
        print('x.T dot x: {}'.format(c_3))
        # update counter
        counter+=1
    return x,pi,z
```

Test function:

```
import numpy as np
# define the qudratic problem
c=np.array([0,0,0])
Q=np.array([[0.02778,0.00387,0.00021],[0.00387,0.01112,-0.0002],[
0.00021,-0.0002,0.00115]])
A=np.array([[0.1073,0.0737,0.0627],[1,1,1]])
b=np.array([0.0650,1])
# define the initial infeasible solution
x=np.array([1,1,1])
pi=np.array([1,1])
z=np.array([1,1,1])
x_optimal,pi_optimal,z_optimal=primal_dual_interior_point(c,Q,A,b
,x,pi,z,0.95)
```

Appendix B

Output of each iteration:

```
=====Iteration0=====
x: [1 1 1]
pi: [1 1]
z: [1 1 1]
tau: -
primal problem: 0.023905
dual problem: 1.0410949999999999
Residual of primal: 2.0079675520286675
Residual of dual: 3.5772208141656563
x.T dot x: 3
=====Iteration1=====
x: [0.05      0.551804   0.62333086]
pi: [-6.81522183 -0.04482545]
z: [1.01338108 0.7850784  0.70481909]
tau: 0.005894069989513771
primal problem: 0.0019956154242216344
dual problem: -0.48981048232987884
Residual of primal: 0.22603174811563292
Residual of dual: 0.4026785558385087
x.T dot x: 0.9232139443968299
=====Iteration2=====
x: [0.0025    0.50919873 0.61635703]
pi: [-20.16128226   1.02286603]
z: [1.27549507 0.60039681 0.3738499 ]
tau: 0.046299944505073375
primal problem: 0.001602622878477464
dual problem: -0.2892199388894904
Residual of primal: 0.12856590827897474
Residual of dual: 0.2290418700357039
x.T dot x: 0.5393350479333184
=====Iteration3=====
x: [1.25000000e-04 3.44997068e-01 7.11183476e-01]
pi: [-45.90249572   2.90202283]
z: [2.08323231 0.54264975 0.03485061]
tau: 0.01629424841263047
primal problem: 0.0009037069012638179
dual problem: -0.08254310281537955
Residual of primal: 0.056529852805135124
Residual of dual: 0.1007086822055196
x.T dot x: 0.2122581533496141
=====Iteration4=====
x: [5.55787214e-04 2.34345151e-01 7.76453087e-01]
```

```
pi: [-35.13799665   2.21395942]
z: [1.56921533 0.38985217 0.00174253]
tau: 0.01753409849697006
primal problem: 0.0006162051484637589
dual problem: -0.07062657082799122
Residual of primal: 0.011399256889168196
Residual of dual: 0.020307927271412913
x.T dot x: 0.09358510909277533
=====Iteration5=====
x: [5.24183749e-04 2.08462037e-01 7.91631453e-01]
pi: [-9.17933441  0.57696261]
z: [4.09608493e-01 1.02351999e-01 8.71265233e-05]
tau: 0.0016046036806465882
primal problem: 0.0005694677262191366
dual problem: -0.020263590256899396
Residual of primal: 0.0006201341514880303
Residual of dual: 0.0011047772121793105
x.T dot x: 0.02162018839573776
=====Iteration6=====
x: [5.20649138e-04 2.07057629e-01 7.92453800e-01]
pi: [-0.4928369  0.0317996]
z: [2.20972772e-02 6.70150534e-03 4.35632616e-06]
tau: 1.7083048286599498e-06
primal problem: 0.0005671542407902004
dual problem: -0.0008019511996499861
Residual of primal: 3.2205319511149676e-05
Residual of dual: 5.7374203664394875e-05
x.T dot x: 0.0014025549190781544
=====Iteration7=====
x: [6.00284476e-04 2.06674214e-01 7.92732595e-01]
pi: [-0.02216286  0.00226355]
z: [1.10486386e-03 1.51914806e-03 3.80579177e-06]
tau: 0.006193037954916246
primal problem: 0.000566653064275297
dual problem: 0.00025631218477287633
Residual of primal: 7.1219037850719995e-06
Residual of dual: 1.2687766010303139e-05
x.T dot x: 0.00031764893918182597
=====Iteration8=====
x: [0.03386353 0.07178969 0.89434678]
pi: [-0.13819711  0.00961943]
z: [6.61548860e-03 1.31618183e-03 6.67819303e-05]
tau: 0.5038995274998375
primal problem: 0.0005074275977462009
```

```
dual problem: 0.00012918931662751429
Residual of primal: 0.0
Residual of dual: 3.6707339412293e-18
x.T dot x: 0.0003782382811186882
=====Iteration9=====
x: [0.03342589 0.07356413 0.89300998]
pi: [-0.02757912  0.00273556]
z: [1.62447465e-03 6.58090914e-05 1.29174972e-05]
tau: 0.0013175325953778468
primal problem: 0.0005067973790760373
dual problem: 0.00043612123124838354
Residual of primal: 0.0
Residual of dual: 2.6090726772098588e-18
x.T dot x: 7.067614782765513e-05
=====Iteration10=====
x: [0.03079954 0.08421276 0.8849877 ]
pi: [-0.01150857  0.00172353]
z: [8.78698054e-04 3.29045457e-06 5.41482773e-06]
tau: 0.021155361147097925
primal problem: 0.0005038046592752499
dual problem: 0.00047167200529676
Residual of primal: 0.0
Residual of dual: 8.481439493802682e-19
x.T dot x: 3.213265397849015e-05
=====Iteration11=====
x: [0.02733624 0.09825487 0.87440888]
pi: [0.00290613 0.00080916]
z: [2.02287777e-04 1.64522729e-07 2.88166410e-07]
tau: 0.011344229659299319
primal problem: 0.0005019268608487994
dual problem: 0.0004961289328350118
Residual of primal: 0.0
Residual of dual: 6.590379024195219e-19
x.T dot x: 5.79792801378745e-06
=====Iteration12=====
x: [0.02646032 0.10180632 0.87173335]
pi: [0.00658834 0.00057457]
z: [3.06212358e-05 8.22613643e-09 2.76272305e-08]
tau: 0.00034754163950934146
primal problem: 0.0005018247405877753
dual problem: 0.0005009895717107544
Residual of primal: 0.0
Residual of dual: 2.214368944449765e-19
x.T dot x: 8.351688770211488e-07
```

```
=====Iteration13=====
x: [0.02631524 0.10239458 0.87129018]
pi: [0.00719818 0.0005357 ]
z: [2.20734779e-06 4.11306821e-10 1.71719962e-09]
tau: 1.6254682940515944e-06
primal problem: 0.0005018223551552063
dual problem: 0.0005017627299759822
Residual of primal: 0.0
Residual of dual: 3.204491494080824e-19
x.T dot x: 5.962517922405285e-08
=====Iteration14=====
x: [0.02630455 0.10243791 0.87125754]
pi: [0.00724311 0.00053284]
z: [1.14029064e-07 2.05653411e-11 8.74423751e-11]
tau: 6.428991229583575e-10
primal problem: 0.0005018223427293778
dual problem: 0.0005018192649545963
Residual of primal: 0.0
Residual of dual: 4.824168370506119e-19
x.T dot x: 3.077774781534848e-09
```