Lab07 - optimalization exercises, Pawel Drapiewski 17.04.2018 r.

Table of Contents

1st excercise from Optimalization_problems.pdf

```
%          y
%   .---------------.
% x |               |
%  /\/\/\/\/\/\/\/\/\
%  /\/\ River /\/\/\/\
%
% Fence length = 2400

% solving the problem using gp option
clear all;
close all;

cvx_begin gp
variables  x y
maximize x * y
subject to
    2 * x + y <= 2400
cvx_end

fprintf('x = %4.2f, y = %4.2f \n', x, y)
fprintf('GP solution %4.2f \n', x * y)


% this problem using logaritms
clear all;
close all;

cvx_begin
```

```
variables  a b
maximize a + b
subject to
    log(2 * exp(a) + exp(b)) <= log(2400)

cvx_end

x = exp(a);
y = exp(b);

fprintf('x = %4.2f, y = %4.2f \n', x, y)
fprintf('Logarithm solution %4.2f \n', x * y)

% Result
% In both cases the results are good


Successive approximation method to be employed.
   SDPT3 will be called several times to refine the solution.
   Original size: 10 variables, 3 equality constraints
   2 exponentials add 16 variables, 10 equality constraints
-----------------------------------------------------------------
 Cones  |             Errors              |
Mov/Act | Centering  Exp cone   Poly cone | Status
--------+---------------------------------+---------
  2/  2 | 6.133e-01  2.918e-02  0.000e+00 | Solved
  2/  2 | 7.273e-02  3.967e-04  0.000e+00 | Solved
  2/  2 | 6.470e-03  3.127e-06  0.000e+00 | Solved
  2/  2 | 5.528e-04  2.267e-08  0.000e+00 | Solved
  0/  2 | 4.705e-05  1.171e-12  0.000e+00 | Solved
-----------------------------------------------------------------
Status: Solved
Optimal value (cvx_optval): +720000

x = 600.00, y = 1200.00
GP solution 720000.00
CVX Warning:
   Models involving "log" or other functions in the log, exp, and
 entropy
   family are solved using an experimental successive approximation
 method.
   This method is slower and less reliable than the method CVX employs
 for
   other models. Please see the section of the user's guide entitled
       <a href="file:///C:\Users\drape\OneDrive\Documents\MATLAB
\Teoria_optymalizacji\cvx\doc\advanced.html#the-successive-
approximation-method">The successive approximation method</a>
   for more details about the approach, and for instructions on how to
   suppress this warning message in the future.

Successive approximation method to be employed.
   SDPT3 will be called several times to refine the solution.
   Original size: 9 variables, 3 equality constraints
   2 exponentials add 16 variables, 10 equality constraints
```

```
------------------------------------------------------------------
 Cones  |             Errors              |
Mov/Act | Centering  Exp cone   Poly cone | Status
--------+---------------------------------+---------
  2/  2 | 6.133e-01  2.918e-02  0.000e+00 | Solved
  2/  2 | 7.273e-02  3.967e-04  0.000e+00 | Solved
  2/  2 | 6.470e-03  3.127e-06  0.000e+00 | Solved
  2/  2 | 5.527e-04  2.254e-08  0.000e+00 | Solved
  0/  0 | 4.464e-05  0.000e+00  0.000e+00 | Solved
------------------------------------------------------------------
Status: Solved
Optimal value (cvx_optval): +13.487

x = 600.00, y = 1200.00
Logarithm solution 720000.00
```

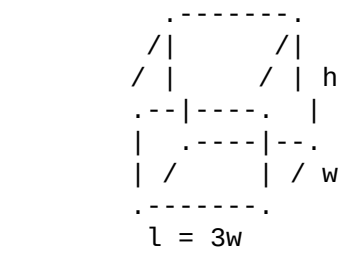
2nd excercise from Optimalization_problem-s.pdf

```
% ################
% ################ - building
% |              |
% |              | y
%  --------------  - fence
%         x
%
% Fence is 500 feet long
% Problem: Find the largest area

% Solution using gp
clear, close all

cvx_begin gp
variables x y
maximize x * y
subject to
    2 * y + x <= 500
cvx_end

fprintf('x = %4.2f, y = %4.2f \n', x, y)
fprintf('GP solution %4.2f \n', x * y)

% Solution using Logarithms
clear, close all

cvx_begin
variables log_x log_y
maximize log_x + log_y
subject to
    log(exp(log_x) + 2*exp(log_y)) <= log(500)
cvx_end
```

```
x = exp(log_x);
y = exp(log_y);

fprintf('x = %4.2f, y = %4.2f \n', x, y)
fprintf('Logarithm solution %4.2f \n', x * y)

% Results
% In both cases the result is the good optimal result
```

```
Successive approximation method to be employed.
   SDPT3 will be called several times to refine the solution.
   Original size: 10 variables, 3 equality constraints
   2 exponentials add 16 variables, 10 equality constraints
-----------------------------------------------------------------
 Cones  |             Errors              |
Mov/Act | Centering  Exp cone   Poly cone | Status
--------+---------------------------------+---------
  2/  2 | 6.133e-01  2.918e-02  0.000e+00 | Solved
  2/  2 | 7.273e-02  3.967e-04  0.000e+00 | Solved
  2/  2 | 6.470e-03  3.127e-06  0.000e+00 | Solved
  2/  2 | 5.528e-04  2.267e-08  0.000e+00 | Solved
  0/  2 | 4.705e-05  1.171e-12  0.000e+00 | Solved
-----------------------------------------------------------------
Status: Solved
Optimal value (cvx_optval): +31250

x = 250.00, y = 125.00
GP solution 31250.00

Successive approximation method to be employed.
   SDPT3 will be called several times to refine the solution.
   Original size: 9 variables, 3 equality constraints
   2 exponentials add 16 variables, 10 equality constraints
-----------------------------------------------------------------
 Cones  |             Errors              |
Mov/Act | Centering  Exp cone   Poly cone | Status
--------+---------------------------------+---------
  2/  2 | 6.133e-01  2.918e-02  0.000e+00 | Solved
  2/  2 | 7.273e-02  3.967e-04  0.000e+00 | Solved
  2/  2 | 6.470e-03  3.127e-06  0.000e+00 | Solved
  2/  2 | 5.527e-04  2.254e-08  0.000e+00 | Solved
  0/  0 | 4.464e-05  0.000e+00  0.000e+00 | Solved
-----------------------------------------------------------------
Status: Solved
Optimal value (cvx_optval): +10.3498

x = 250.00, y = 125.00
Logarithm solution 31250.00
```

3rd excercise from Optimalization_problem-s.pdf

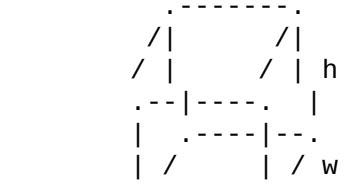


top and bottom material cost = $10 per ft^2 sides material cost = $6 per ft^2

```
% GP solution
clear, close all

cvx_begin gp
variables w
minimize 60 * w * w  + 800 / w
subject to
    w <= 1 % because lhw <= 50 we can write as 50w^4 < 50, so w <= 1
cvx_end

h = 50 / 3 * w * w;
l = 3 * w;

fprintf('w = %4.2f, h = %4.2f, l = %4.2f \n', w, h, l)
fprintf('GP solution %4.2f \n', 60 * w^2 + 48 * w * h)
fprintf('Box volume %4.2f \n', l * h * w)

% Logarithm solution
clear, close all

cvx_begin
variables log_w
minimize 60 * exp(log_w) * exp(log_w)  + 800 / exp(log_w)
subject to
    exp(log_w) <= log(1)
cvx_end

w = exp(log_w);
h = 50 / ( 3 * w * w);
l = 3 * w;

fprintf('w = %4.2f, h = %4.2f, l = %4.2f \n', w, h, l)
fprintf('Logarithm solution %4.2f \n', 60 * w^2 + 48 * w * h)
fprintf('Box volume %4.2f \n', l * h * w)

% Results
% The Logarithm optimalization found good value and GP not, not even
 close
```

```
% enough


Successive approximation method to be employed.
   SDPT3 will be called several times to refine the solution.
   Original size: 10 variables, 4 equality constraints
   2 exponentials add 16 variables, 10 equality constraints
-----------------------------------------------------------------
 Cones  |             Errors              |
Mov/Act | Centering  Exp cone   Poly cone | Status
--------+---------------------------------+---------
  2/  2 | 2.753e+00  5.599e-01  0.000e+00 | Solved
  2/  2 | 8.717e-02  5.065e-04  0.000e+00 | Solved
  2/  2 | 2.987e-03  5.867e-07  0.000e+00 | Solved
  0/  1 | 1.532e-04  9.313e-10  0.000e+00 | Solved
-----------------------------------------------------------------
Status: Solved
Optimal value (cvx_optval): +860

w = 1.00, h = 16.67, l = 3.00
GP solution 860.00
Box volume 50.00

Successive approximation method to be employed.
   SDPT3 will be called several times to refine the solution.
   Original size: 10 variables, 4 equality constraints
   2 exponentials add 16 variables, 10 equality constraints
-----------------------------------------------------------------
 Cones  |             Errors              |
Mov/Act | Centering  Exp cone   Poly cone | Status
--------+---------------------------------+---------
  2/  2 | 1.048e+00  8.094e-02  0.000e+00 | Solved
  2/  2 | 5.885e-02  2.501e-04  0.000e+00 | Solved
  2/  2 | 3.838e-03  1.033e-06  0.000e+00 | Solved
  0/  2 | 2.576e-04  4.745e-09  0.000e+00 | Solved
-----------------------------------------------------------------
Status: Solved
Optimal value (cvx_optval): +637.595

w = 1.88, h = 4.71, l = 5.65
Logarithm solution 637.60
Box volume 50.00
```

4th excercise from Optimalization_problem-s.pdf



```
        .--------.
         l = w
```

We have only 10 m^2 of material

```
% Solution
clear, close all

cvx_begin gp
variables w h
maximize w * w * h  % what is V = lwh
subject to
    2 * w * w + 4 * w * h <= 10
cvx_end


h = (5 - w * w) / (2 * w);
l = w;

fprintf('w = %4.2f, h = %4.2f, l = %4.2f \n', w, h, l)
fprintf('Box volume %4.2f \n', l * h * w)
```

```
Successive approximation method to be employed.
   SDPT3 will be called several times to refine the solution.
   Original size: 13 variables, 3 equality constraints
   2 exponentials add 16 variables, 10 equality constraints
-----------------------------------------------------------------
 Cones  |             Errors              |
Mov/Act | Centering  Exp cone   Poly cone | Status
--------+---------------------------------+---------
  2/  2 | 1.030e+00  8.095e-02  0.000e+00 | Solved
  2/  2 | 6.350e-02  2.917e-04  0.000e+00 | Solved
  2/  2 | 5.248e-03  2.044e-06  0.000e+00 | Solved
  0/  2 | 3.890e-04  9.348e-09  0.000e+00 | Solved
-----------------------------------------------------------------
Status: Solved
Optimal value (cvx_optval): +2.15166

w = 1.29, h = 1.29, l = 1.29
Box volume 2.15
```

5th excercise from Optimalization_problem-s.pdf

Make cylider that hold 1.5 liters of liquid. Minimize the dimenstions

```
% Solution
clear, close all

cvx_begin gp
variables h r
minimize 2 * pi * r * r + 2 * pi * r * h
```

```
subject to
   pi * r * r * h == 1500
cvx_end

fprintf('h = %4.2f, r = %4.2f \n', h, r)
fprintf('Needed material [m^2]: %4.2f \n', 2 * pi * r * r + 2 * pi * r
 * h)
fprintf('Valume of the cylider needed: %4.2f \n',  pi * r * r * h)


Successive approximation method to be employed.
   SDPT3 will be called several times to refine the solution.
   Original size: 12 variables, 3 equality constraints
   2 exponentials add 16 variables, 10 equality constraints
-----------------------------------------------------------------
 Cones  |             Errors              |
Mov/Act | Centering  Exp cone   Poly cone | Status
--------+---------------------------------+---------
  2/  2 | 1.048e+00  8.095e-02  0.000e+00 | Solved
  2/  2 | 5.885e-02  2.501e-04  0.000e+00 | Solved
  2/  2 | 3.840e-03  1.032e-06  0.000e+00 | Solved
  0/  2 | 2.577e-04  4.672e-09  0.000e+00 | Solved
-----------------------------------------------------------------
Status: Solved
Optimal value (cvx_optval): +725.396

h = 12.41, r = 6.20
Needed material [m^2]: 725.40
Valume of the cylider needed: 1500.00
```

6th excercise from Optimalization_problem-s.pdf

```
clear, close all;

cvx_begin gp
variables x y h
maximize x * y * h
subject to
    x + 2 * h <= 14
    y + 2 * h <= 10
cvx_end

fprintf('x=%2.4f, y=%2.4f, h=%2.4f \n', x, y, h)
fprintf('Result: %2.4f \n', x * y * h)


Successive approximation method to be employed.
   SDPT3 will be called several times to refine the solution.
   Original size: 19 variables, 7 equality constraints
   4 exponentials add 32 variables, 20 equality constraints
-----------------------------------------------------------------
 Cones  |             Errors              |
```

```
Mov/Act | Centering  Exp cone   Poly cone | Status
--------+---------------------------------+---------
  4/  4 | 1.230e+00  1.146e-01  0.000e+00 | Solved
  4/  4 | 6.665e-02  3.307e-04  0.000e+00 | Solved
  4/  4 | 5.780e-03  2.481e-06  0.000e+00 | Solved
  0/  4 | 4.386e-04  1.402e-08  0.000e+00 | Solved
-----------------------------------------------------------------
Status: Solved
Optimal value (cvx_optval): +120.164

x=10.1633, y=6.1633, h=1.9184
Result: 120.1644
```

7th excercise from Optimalization_problem-s.pdf

Print sheet: .----------. | 2 inch | - outter rectangle is template | - - - - - | - inner rectangle is print area 1 1 | ____ | |1.5 inch | .----------.

Maximize the printed area Whole template have 200 in^2 area.

```
% Solution
clear, close all;

cvx_begin
variables log_with_w log_with_h
maximize log_with_w + log_with_h % orginal: (w - 2) * (h - 3.5)
subject to
   log((exp(log_with_w) + 2)  * (exp(log_with_h) + 3.5)) <= log(200);
cvx_end

w = exp(log_with_w) + 2;
h = exp(log_with_h) + 3.5;

fprintf('w=%2.4f, h=%2.4f \n', w, h)
fprintf('Maximized printed area: %2.4f \n',(w - 2) * (h - 3.5))
fprintf('Template area %2.4f \n', w * h);


Successive approximation method to be employed.
   SDPT3 will be called several times to refine the solution.
   Original size: 17 variables, 7 equality constraints
   4 exponentials add 32 variables, 20 equality constraints
-----------------------------------------------------------------
 Cones  |             Errors              |
Mov/Act | Centering  Exp cone   Poly cone | Status
--------+---------------------------------+---------
  4/  4 | 1.654e+00  2.048e-01  0.000e+00 | Solved
  4/  4 | 5.087e-02  1.898e-04  0.000e+00 | Solved
  4/  4 | 3.978e-03  1.157e-06  0.000e+00 | Solved
  0/  2 | 3.012e-04  6.567e-09  0.000e+00 | Solved
-----------------------------------------------------------------
Status: Solved
```

```
Optimal value (cvx_optval): +4.88407

w=10.6904, h=18.7083
Maximized printed area: 132.1669
Template area 200.0000
```

8th excercise from Optimalization_problem-s.pdf

```
clear, close all;

cvx_begin
variables r h
maximize 12 * r - (2 + pi/2) * r^2
subject to
   2*h + 2*r + pi*r == 12;
cvx_end

fprintf('r=%2.4f, h=%2.4f \n', r, h)
fprintf('Window surface: %2.4f \n', 12 * r - (2 + pi/2) * r^2)


Calling SDPT3 4.0: 3 variables, 1 equality constraints
------------------------------------------------------------

 num. of constraints =  1
 dim. of sdp    var  =  2,   num. of sdp  blk  =  1
*****************************************************************
   SDPT3: Infeasible path-following algorithms
*****************************************************************
 version  predcorr  gam  expon  scale_data
   HKM      1      0.000   1        0
it pstep dstep pinfeas dinfeas  gap      prim-obj      dual-obj
 cputime
-----------------------------------------------------------------
 0|0.000|0.000|4.5e+00|1.5e+00|2.0e+02| 3.570796e+01  0.000000e+00|
 0:0:00| chol  1  1
 1|0.868|1.000|5.9e-01|1.4e-02|9.4e+00|-1.720249e+01 -1.169740e+01|
 0:0:00| chol  1  1
 2|0.992|1.000|4.8e-03|1.4e-03|6.6e-01|-1.012252e+01 -1.064597e+01|
 0:0:00| chol  1  1
 3|0.982|0.988|8.6e-05|1.8e-04|9.5e-03|-1.008255e+01 -1.008518e+01|
 0:0:00| chol  1  1
 4|0.960|0.982|3.5e-06|1.8e-05|2.2e-04|-1.008182e+01 -1.008148e+01|
 0:0:00| chol  1  1
 5|0.941|0.974|2.1e-07|4.9e-07|9.3e-06|-1.008179e+01 -1.008178e+01|
 0:0:00| chol  1  1
 6|0.989|0.988|2.3e-09|7.4e-09|4.8e-07|-1.008178e+01 -1.008178e+01|
 0:0:00| chol  1  1
 7|1.000|1.000|0.0e+00|1.8e-10|5.5e-08|-1.008178e+01 -1.008178e+01|
 0:0:00|
  stop: max(relative gap, infeasibilities) < 1.49e-08
```

```
-----------------------------------------------------------------
 number of iterations   =  7
 primal objective value = -1.00817847e+01
 dual   objective value = -1.00817848e+01
 gap := trace(XZ)       = 5.53e-08
 relative gap           = 2.61e-09
 actual relative gap    = 2.37e-09
 rel. primal infeas (scaled problem)   = 0.00e+00
 rel. dual     "        "       "      = 1.82e-10
 rel. primal infeas (unscaled problem) = 0.00e+00
 rel. dual     "        "       "      = 0.00e+00
 norm(X), norm(y), norm(Z) = 3.8e+00, 1.0e+01, 1.4e+01
 norm(A), norm(b), norm(C) = 2.0e+00, 2.0e+00, 1.0e+01
 Total CPU time (secs)  = 0.39
 CPU time per iteration = 0.06
 termination code       =  0
 DIMACS: 0.0e+00  0.0e+00  2.7e-10  0.0e+00  2.4e-09  2.6e-09
-----------------------------------------------------------------

------------------------------------------------------------
Status: Solved
Optimal value (cvx_optval): +10.0818

r=1.6803, h=1.6802
Window surface: 10.0818
```

9th excercise from Optimalization_problem-s.pdf

```
% The biggest possible rectangle drawn into circle of radius 4
% x, y - sides of the rectangle

% Solution
clear, close all;

cvx_begin
variables v
maximize 256 * v - 16 * v * v
subject to
    - v <= 0
cvx_end

u = 16 - v;
x = sqrt(u);
y = sqrt(v);

fprintf('x=%2.4f, y=%2.4f \n', x, y)
fprintf('Rectnalge area: %2.4f \n', 2 * x + 2 * y)


Calling SDPT3 4.0: 4 variables, 2 equality constraints
   For improved efficiency, SDPT3 is solving the dual problem.
```

```
------------------------------------------------------------

 num. of constraints =  2
 dim. of sdp    var  =  2,   num. of sdp  blk  =  1
 dim. of linear var  =  1
*****************************************************************
   SDPT3: Infeasible path-following algorithms
*****************************************************************
 version  predcorr  gam  expon  scale_data
   HKM      1      0.000   1        0
it pstep dstep pinfeas dinfeas  gap      prim-obj      dual-obj
 cputime
-----------------------------------------------------------------
 0|0.000|0.000|1.7e+00|8.4e+00|5.5e+03| 2.129058e+02  0.000000e+00|
 0:0:00| chol  1  1
 1|0.804|0.819|3.3e-01|1.6e+00|5.0e+02| 2.619005e+02  1.224143e+03|
 0:0:00| chol  1  1
 2|0.622|0.630|1.2e-01|5.9e-01|2.0e+02| 5.178792e+02  1.252930e+03|
 0:0:00| chol  1  1
 3|0.669|0.631|4.1e-02|2.2e-01|9.1e+01| 7.923507e+02  1.180887e+03|
 0:0:00| chol  1  1
 4|0.818|0.688|7.5e-03|6.9e-02|2.9e+01| 9.806937e+02  1.092807e+03|
 0:0:00| chol  1  1
 5|1.000|0.998|4.6e-08|7.1e-04|4.8e+00| 1.027793e+03  1.023820e+03|
 0:0:00| chol  1  1
 6|0.970|1.000|3.4e-09|7.0e-05|1.1e-01| 1.024091e+03  1.024060e+03|
 0:0:00| chol  1  1
 7|0.939|1.000|1.1e-09|6.8e-10|1.0e-02| 1.024006e+03  1.023996e+03|
 0:0:00| chol  1  1
 8|0.980|0.982|1.8e-10|2.3e-10|2.4e-04| 1.024000e+03  1.024000e+03|
 0:0:00| chol  1  1
 9|0.988|0.989|3.6e-12|3.8e-11|2.8e-06| 1.024000e+03  1.024000e+03|
 0:0:00|
  stop: max(relative gap, infeasibilities) < 1.49e-08
-----------------------------------------------------------------
 number of iterations   =  9
 primal objective value =  1.02400000e+03
 dual   objective value =  1.02400000e+03
 gap := trace(XZ)       = 2.83e-06
 relative gap           = 1.38e-09
 actual relative gap    = 1.39e-09
 rel. primal infeas (scaled problem)   = 3.63e-12
 rel. dual     "        "       "      = 3.81e-11
 rel. primal infeas (unscaled problem) = 0.00e+00
 rel. dual     "        "       "      = 0.00e+00
 norm(X), norm(y), norm(Z) = 1.0e+03, 6.4e+01, 6.5e+01
 norm(A), norm(b), norm(C) = 3.0e+00, 2.6e+02, 2.0e+00
 Total CPU time (secs)  = 0.29
 CPU time per iteration = 0.03
 termination code       =  0
 DIMACS: 3.6e-12  0.0e+00  3.8e-11  0.0e+00  1.4e-09  1.4e-09
-----------------------------------------------------------------

------------------------------------------------------------
```

```
Status: Solved
Optimal value (cvx_optval): +1024

x=2.8284, y=2.8284
Rectnalge area: 11.3137
```

10th excercise from Optimalization_problem-s.pdf

```
% Solution
clear, close all;

cvx_solver sdpt3
cvx_begin
variables y
minimize y * y - 3 * y + 3
subject to
    - y <= 0
cvx_end

x_p2 = y - 1;
x = sqrt(x_p2);
fprintf('x=%2.4f \n', x)


Calling SDPT3 4.0: 4 variables, 2 equality constraints
   For improved efficiency, SDPT3 is solving the dual problem.
------------------------------------------------------------

 num. of constraints =  2
 dim. of sdp    var  =  2,   num. of sdp  blk  =  1
 dim. of linear var  =  1
*****************************************************************
   SDPT3: Infeasible path-following algorithms
*****************************************************************
 version  predcorr  gam  expon  scale_data
   HKM      1      0.000   1        0
it pstep dstep pinfeas dinfeas  gap      prim-obj      dual-obj
 cputime
-----------------------------------------------------------------
 0|0.000|0.000|3.8e+00|8.4e+00|3.0e+02| 1.000000e+01  0.000000e+00|
 0:0:00| chol  1  1
 1|0.962|0.970|1.4e-01|3.4e-01|2.0e+01| 9.814211e+00  6.526019e-01|
 0:0:00| chol  1  1
 2|0.956|0.953|6.4e-03|2.4e-02|2.4e+00| 4.173971e+00  2.035416e+00|
 0:0:00| chol  1  1
 3|1.000|1.000|4.5e-06|2.1e-03|1.7e+00| 3.402998e+00  1.695028e+00|
 0:0:00| chol  1  1
 4|0.987|0.981|6.7e-08|1.3e-04|2.6e-02| 2.267410e+00  2.242150e+00|
 0:0:00| chol  1  1
 5|0.983|0.994|3.9e-10|9.4e-06|3.5e-04| 2.250230e+00  2.249917e+00|
 0:0:00| chol  1  1
```

```
 6|0.940|1.000|4.9e-09|7.8e-11|1.4e-05| 2.250009e+00  2.249995e+00|
 0:0:00| chol  1  1
 7|0.977|1.000|9.2e-10|1.2e-10|4.5e-07| 2.250000e+00  2.250000e+00|
 0:0:00| chol  2  2
 8|1.000|1.000|2.4e-11|1.8e-10|1.1e-08| 2.250000e+00  2.250000e+00|
 0:0:00|
  stop: max(relative gap, infeasibilities) < 1.49e-08
------------------------------------------------------------------------
 number of iterations   =  8
 primal objective value =  2.25000001e+00
 dual   objective value =  2.25000000e+00
 gap := trace(XZ)       = 1.11e-08
 relative gap           = 2.01e-09
 actual relative gap    = 1.84e-09
 rel. primal infeas (scaled problem)   = 2.41e-11
 rel. dual     "        "       "      = 1.76e-10
 rel. primal infeas (unscaled problem) = 0.00e+00
 rel. dual     "        "       "      = 0.00e+00
 norm(X), norm(y), norm(Z) = 3.3e+00, 2.7e+00, 3.6e+00
 norm(A), norm(b), norm(C) = 3.0e+00, 4.2e+00, 2.0e+00
 Total CPU time (secs)  = 0.21
 CPU time per iteration = 0.03
 termination code       =  0
 DIMACS: 2.5e-11  0.0e+00  1.8e-10  0.0e+00  1.8e-09  2.0e-09
------------------------------------------------------------------------

------------------------------------------------------------
Status: Solved
Optimal value (cvx_optval): +0.75

x=0.7071
```

11th excercise from Optimalization_problem-s.pdf

```
% River - the fastest way to reach point on the opposite bank

% Solution
clear, close all;

cvx_begin
variables x
minimize (norm([x 9]) / 6 - x/8)
subject to
    - x <= 0
cvx_end

fprintf('x=%2.4f \n', x)
```

```
Calling SDPT3 4.0: 4 variables, 2 equality constraints
   For improved efficiency, SDPT3 is solving the dual problem.
```

```
-----------------------------------------------------------

 num. of constraints =  2
 dim. of socp   var  =  3,   num. of socp blk  =  1
 dim. of linear var  =  1
*****************************************************************
   SDPT3: Infeasible path-following algorithms
*****************************************************************
 version  predcorr  gam  expon  scale_data
    NT      1      0.000   1        0
it pstep dstep pinfeas dinfeas  gap      prim-obj      dual-obj
 cputime
-----------------------------------------------------------------
 0|0.000|0.000|8.5e+00|1.7e+00|1.2e+02| 0.000000e+00  0.000000e+00|
 0:0:00| chol  1  1
 1|0.762|1.000|2.0e+00|1.4e-02|3.7e+01|-2.058565e+00 -4.003185e+00|
 0:0:00| chol  1  1
 2|0.988|1.000|2.4e-02|1.4e-03|1.9e+00|-8.144566e-01 -2.341546e+00|
 0:0:00| chol  1  1
 3|1.000|0.960|6.3e-09|5.0e-03|7.4e-02|-9.526497e-01 -1.020554e+00|
 0:0:00| chol  1  1
 4|0.908|1.000|1.1e-08|1.4e-05|3.9e-03|-9.904720e-01 -9.943559e-01|
 0:0:00| chol  1  1
 5|0.920|1.000|3.3e-09|1.4e-06|1.5e-04|-9.920985e-01 -9.922462e-01|
 0:0:00| chol  1  1
 6|0.919|0.992|5.1e-09|1.1e-08|6.3e-06|-9.921535e-01 -9.921597e-01|
 0:0:00| chol  1  1
 7|0.948|0.987|6.3e-09|1.1e-09|2.2e-07|-9.921567e-01 -9.921568e-01|
 0:0:00| chol  1  2
 8|1.000|1.000|1.3e-09|6.0e-10|7.8e-09|-9.921568e-01 -9.921567e-01|
 0:0:00|
  stop: max(relative gap, infeasibilities) < 1.49e-08
-----------------------------------------------------------------
 number of iterations   =  8
 primal objective value = -9.92156759e-01
 dual   objective value = -9.92156744e-01
 gap := trace(XZ)       = 7.75e-09
 relative gap           = 2.60e-09
 actual relative gap    = -5.23e-09
 rel. primal infeas (scaled problem)   = 1.31e-09
 rel. dual     "        "       "      = 5.98e-10
 rel. primal infeas (unscaled problem) = 0.00e+00
 rel. dual     "        "       "      = 0.00e+00
 norm(X), norm(y), norm(Z) = 2.4e-01, 1.7e+01, 2.2e+01
 norm(A), norm(b), norm(C) = 2.7e+00, 1.2e+00, 1.0e+01
 Total CPU time (secs)  = 0.43
 CPU time per iteration = 0.05
 termination code       =  0
 DIMACS: 1.4e-09  0.0e+00  6.0e-10  0.0e+00  -5.2e-09  2.6e-09
-----------------------------------------------------------------

-----------------------------------------------------------
Status: Solved
Optimal value (cvx_optval): +0.992157
```

```
x=10.2049
```

Published with MATLAB® R2017b