Maximum volume ellipsoid in intersection of ellipses and polyhedra: Matlab+Yalmip+SeDuMi example

© 2024, Antonio Sala. Universitat Politecnica de Valencia, Spain. All rights reserved.

Presentation in video: https://personales.upv.es/asala/YT/V/elinncEN.html

Objective: illustrating LMIs (semidefinite programming) in geometric problem with ellipses not centered at the origin.

*This code worked with Matlab R2024a+yalmip:R20230609+sedumi:1.3

The region in which we'll fit the ellipse

```
if ~exist('yalmiptest','file') %add toolboxes to path
    addpath("~/bin/MATLAB/tbxmanager/")
    tbxmanager restorepath
end %depends on your installation method/source
```

```
Toolbox "sedumi:1.3:glnxa64" added to the Matlab path.
Toolbox "yalmip:R20230609:all" added to the Matlab path.
Toolbox "mpt:3.2.1:all" added to the Matlab path.
Toolbox "mptdoc:3.0.4:all" added to the Matlab path.
Toolbox "cddmex:1.0.1:glnxa64" added to the Matlab path.
Toolbox "fourier:1.0:glnxa64" added to the Matlab path.
Toolbox "glpkmex:1.0:glnxa64" added to the Matlab path.
Toolbox "hysdel:2.0.6:glnxa64" added to the Matlab path.
Toolbox "lcp:1.0.3:glnxa64" added to the Matlab path.
Toolbox "espresso:1.0:glnxa64" added to the Matlab path.
```

An ellipsoid $(x-c_1)^T Q_1^{-1}(x-c) \leq 1$ with:

```
Q1=[3 -1;-1 4]*2;
c1=[2;1];
```

A polyhedron $Ax \leq b$ with:

```
A=[1 -.1; -1 0; 0.1 1; 0.2 -1];
b=[3; 0.1; 3; 2];
Nfaces=size(A,1)
```

```
Nfaces = 4
```

Let us plot the intersection region:

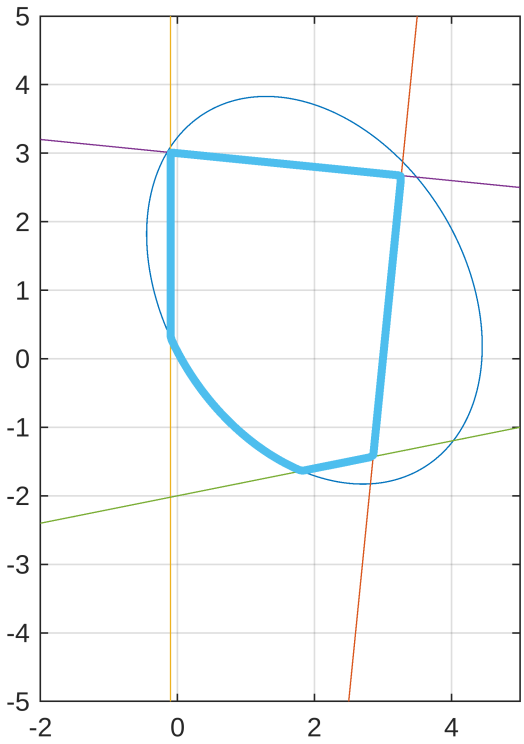
```
Ellipfun=@(x1,x2) 1-([x1;x2]-c1)'*inv(Q1)*([x1;x2]-c1); %>=0, we are in
the ellipsoid
for i=1:Nfaces
    PolyFun{i}=@(x1,x2) -A(i,:)*[x1;x2]+b(i); %>=0 we are at the right
side of each face
end
SetFun=@(x1,x2) min([Ellipfun(x1,x2),PolyFun{1}(x1,x2),PolyFun{2}
(x1,x2),PolyFun{3}(x1,x2),PolyFun{4}(x1,x2)]);
%If the minimum is >=0, all of them are >=0. This is the set boundary.
axx=[-2 5 -5 5];
fimplicit(Ellipfun,axx)
```

```
Warning: MATLAB has disabled some advanced graphics rendering features by switching to software
OpenGL. For more information, click here.
Warning: Function behaves unexpectedly on array inputs. To improve performance, properly
vectorize your function to return an output with the same size and shape as the input arguments.
```

```
hold on
for i=1:Nfaces
    fimplicit(PolyFun{i},axx)
end
grid on
fimplicit(SetFun,axx,LineWidth=3)
```

```
Warning: Function behaves unexpectedly on array inputs. To improve performance, properly
vectorize your function to return an output with the same size and shape as the input arguments.
```

```
hold off, axis equal
```



Finding the maximum-area ellipse in the region

For convenience, let us look for a maximum volume ellipsoid in the form $\|H^{-1}(x-c)\| \leq 1$, which means $(x-c)^T H^{-2}(x-c) \leq 1$... we are assumming symmetric H with no loss of generality $H^T H = Q$ has a symmetric square root (we can get it from eigenvectors and square root of eigenvalues).

```
sdpvar cx cy lam
 c=[cx;cy];

H=sdpvar(2); %symmetric square root of sought ellipsoid
%sdpvar radius  %largest circle
%H=radius*eye(2);
LMIs= H>=0 ; %we with positive square root of Q=H^TH.
```

First non-trivial LMI: target ellipsoid inside the above-defined one, see

https://web.stanford.edu/~boyd/lmibook/lmibook.pdf page 46, (3.17)

```
LMIs=[LMIs;
    [-Q1       c1-c    H; ...
     (c1-c)' lam-1   0 0;...
      H'     [0;0] -lam*eye(2)]<=0 ;
      lam >=0]; %last one is actually redundant, but, well, it's in
Boyd's book
```

Second group of LMIs: target ellipsoid inside polyhedron, face by face, see

https://users.ece.utexas.edu/~cmcaram/EE381V_2012F/Lecture_17_Scribe_Notes.final.pdf page 17-6, equation (17.11):

```
for i=1:Nfaces
    LMIs=[LMIs;
        [b(i)-A(i,:)*c        A(i,:)*H; ...
           H'*A(i,:)' (b(i)-A(i,:)*c)*eye(2)]>=0 ];
end
sol=optimize(LMIs,-geomean(H)) %geometric mean is propotional to
determinant, maximization
```

```
SeDuMi 1.3 by AdvOL, 2005-2008 and Jos F. Sturm, 1998-2003.
Alg = 2: xz-corrector, theta = 0.250, beta = 0.500
eqs m = 10, order n = 26, dim = 85, blocks = 9
nnz(A) = 60 + 0, nnz(ADA) = 68, nnz(L) = 39
 it :     b*y       gap    delta  rate   t/tP*  t/tD*   feas cg cg  prec
  0 :            2.29E+01 0.000
  1 :   2.90E+00 7.33E+00 0.000 0.3207 0.9000 0.9000   2.32  1  1  1.6E+00
  2 :   1.46E+00 1.64E+00 0.000 0.2239 0.9000 0.9000   1.88  1  1  4.0E-01
  3 :   1.75E+00 3.19E-01 0.000 0.1941 0.9000 0.9000   1.11  1  1  1.0E-01
  4 :   1.81E+00 2.80E-02 0.000 0.0878 0.9900 0.9900   0.99  1  1  1.0E-02
  5 :   1.81E+00 5.30E-03 0.000 0.1892 0.9000 0.9000   1.00  1  1  2.0E-03
  6 :   1.81E+00 3.64E-04 0.478 0.0688 0.9900 0.9900   1.00  1  1  1.4E-04
  7 :   1.81E+00 7.47E-05 0.000 0.2052 0.9000 0.9000   1.00  1  1  2.8E-05
  8 :   1.81E+00 6.76E-06 0.335 0.0905 0.9900 0.9900   1.00  1  1  2.6E-06
  9 :   1.81E+00 1.38E-06 0.000 0.2041 0.9000 0.9000   1.00  1  1  5.2E-07
 10 :   1.81E+00 3.15E-07 0.000 0.2279 0.9000 0.9000   1.00  1  1  1.2E-07
 11 :   1.81E+00 5.49E-08 0.000 0.1746 0.9000 0.9000   1.00  2  2  2.1E-08
 12 :   1.81E+00 1.13E-08 0.000 0.2052 0.9000 0.9000   1.00  2  2  4.3E-09
 13 :   1.81E+00 2.41E-09 0.000 0.2141 0.9000 0.9000   1.00  2  2  9.2E-10

iter seconds digits       c*x               b*y
 13      1.1   8.7  1.8057390264e+00  1.8057390231e+00
|Ax-b| =   3.0e-10, [Ay-c]_+ =   3.0E-10, |x|=  2.2e+00, |y|=  4.5e+00

Detailed timing (sec)
   Pre          IPM          Post
2.198E-01    6.548E-01    2.540E-02
Max-norms: ||b||=1, ||c|| = 8,
Cholesky |add|=0, |skip| = 0, ||L.L|| = 1075.94.
sol = struct with fields:
    yalmipversion: '20230622'
    matlabversion: '24.1.0.2653294 (R2024a) Update 5'
       yalmiptime: 2.5898
       solvertime: 0.9884
             info: 'Successfully solved (SeDuMi)'
          problem: 0
```

```
Hval=value(H)
```

```
Hval = 2×2
    1.5339   -0.2319
   -0.2319    2.1608
```

```
cval=value(c)
```

```
cval = 2×1
    1.4513
    0.7158
```

```
fimplicit(SetFun,axx*.8,LineWidth=2)
```

```
Warning: Function behaves unexpectedly on array inputs. To improve performance, properly
vectorize your function to return an output with the same size and shape as the input arguments.
```

```
hold on
EllipfunSOL=@(x1,x2) 1-([x1;x2]-cval)'*inv(Hval^2)*([x1;x2]-cval);
fimplicit(EllipfunSOL,axx*.8,LineWidth=2)
```

```
Warning: Function behaves unexpectedly on array inputs. To improve performance, properly
vectorize your function to return an output with the same size and shape as the input arguments.
```

```
plot(cval(1),cval(2),'x',LineWidth=3)
hold off, grid on, axis equal
legend("Set boundary", "Extremal Ellipse","center",Location="best")
```