

Lasserre hierarchy example

We want to maximise $f:=x+8y$ subject to $g:=1-(x^4+y^4)\geq 0$

```
R.<x, y> = QQ[]
```

```
X = Curve(x^4 + y^4-1)
```

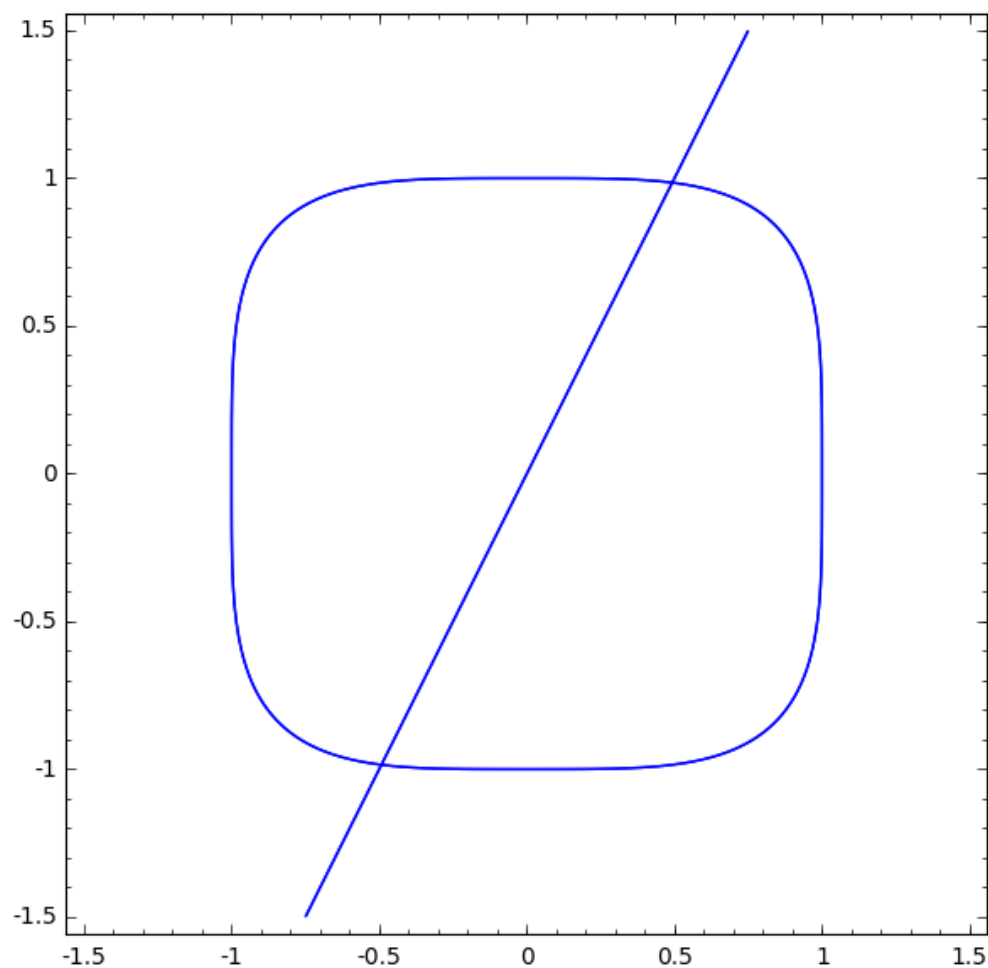
```
Y = Curve(y-2*x)
```

```
g = Graphics()
```

```
g += X.plot((x, -1,1),(y, -1,1),plot_points=1000)
```

```
g += Y.plot((x, -1.5,1.5),(y, -1.5,1.5),plot_points=1000)
```

```
show(g)
```



```
x0=n(17^(-1/4))
```

```
17*x0
```

```
8.37214402859269
```

```
p = SemidefiniteProgram()
```

```
x = p.new_variable()
```

```
(a,b,c,d,e,h,i,j,k,l,m,n,o,q)=  
(x[1],x[2],x[3],x[4],x[5],x[6],x[7],x[8],x[9],x[10],x[11],x[12],x[13],x[14])
```

```
B=matrix(  
[[1,0,0,0,0,0,0],[0,0,0,0,0,0,0],[0,0,0,0,0,0,0],[0,0,0,0,0,0,0],  
[0,0,0,0,0,0,0],[0,0,0,0,0,0,0],[0,0,0,0,0,0,1]]  
)  
+a*matrix(  
[[0,1,0,0,0,0,0],[1,0,0,0,0,0,0],[0,0,0,0,0,0,0],[0,0,0,0,0,0,0],  
[0,0,0,0,0,0,0],[0,0,0,0,0,0,0],[0,0,0,0,0,0,0]]  
)  
+b*matrix(  
[[0,0,1,0,0,0,0],[0,0,0,0,0,0,0],[1,0,0,0,0,0,0],[0,0,0,0,0,0,0],  
[0,0,0,0,0,0,0],[0,0,0,0,0,0,0],[0,0,0,0,0,0,0]]  
)  
+c*matrix(  
[[0,0,0,1,0,0,0],[0,1,0,0,0,0,0],[0,0,0,0,0,0,0],[1,0,0,0,0,0,0],  
[0,0,0,0,0,0,0],[0,0,0,0,0,0,0],[0,0,0,0,0,0,0]]  
)  
+d*matrix(  
[[0,0,0,0,1,0,0],[0,0,1,0,0,0,0],[0,1,0,0,0,0,0],[0,0,0,0,0,0,0],  
[1,0,0,0,0,0,0],[0,0,0,0,0,0,0],[0,0,0,0,0,0,0]]  
)  
+e*matrix(  
[[0,0,0,0,0,1,0],[0,0,0,0,0,0,0],[0,0,1,0,0,0,0],[0,0,0,0,0,0,0],  
[0,0,0,0,0,0,0],[1,0,0,0,0,0,0],[0,0,0,0,0,0,0]]  
)  
+h*matrix(  
[[0,0,0,0,0,0,0],[0,0,0,1,0,0,0],[0,0,0,0,0,0,0],[0,1,0,0,0,0,0],  
[0,0,0,0,0,0,0],[0,0,0,0,0,0,0],[0,0,0,0,0,0,0]]  
)  
+i*matrix(  
[[0,0,0,0,0,0,0],[0,0,0,0,1,0,0],[0,0,0,1,0,0,0],[0,0,1,0,0,0,0],  
[0,1,0,0,0,0,0],[0,0,0,0,0,0,0],[0,0,0,0,0,0,0]]  
)  
+j*matrix(  
[[0,0,0,0,0,0,0],[0,0,0,0,0,1,0],[0,0,0,0,1,0,0],[0,0,0,0,0,0,0],  
[0,0,1,0,0,0,0],[0,1,0,0,0,0,0],[0,0,0,0,0,0,0]]  
)  
+k*matrix(  
[[0,0,0,0,0,0,0],[0,0,0,0,0,0,0],[0,0,0,0,0,1,0],[0,0,0,0,0,0,0],  
[0,0,0,0,0,0,0],[0,0,1,0,0,0,0],[0,0,0,0,0,0,0]]  
)  
+l*matrix(  
[[0,0,0,0,0,0,0],[0,0,0,0,0,0,0],[0,0,0,0,0,0,0],[0,0,0,1,0,0,0],  
[0,0,0,0,0,0,0],[0,0,0,0,0,0,0],[0,0,0,0,0,0,-1]]  
)  
+m*matrix(  
[[0,0,0,0,0,0,0],[0,0,0,0,0,0,0],[0,0,0,0,0,0,0],[0,0,0,0,1,0,0],  
[0,0,0,1,0,0,0],[0,0,0,0,0,0,0],[0,0,0,0,0,0,0]]  
)  
+n*matrix(  
[[0,0,0,0,0,0,0],[0,0,0,0,0,0,0],[0,0,0,0,0,0,0],[0,0,0,0,0,1,0],  
[0,0,0,0,1,0,0],[0,0,0,1,0,0,0],[0,0,0,0,0,0,0]]  
)  
+o*matrix(  
[[0,0,0,0,0,0,0],[0,0,0,0,0,0,0],[0,0,0,0,0,0,0],[0,0,0,0,0,0,0],  
[0,0,0,0,0,1,0],[0,0,0,0,1,0,0],[0,0,0,0,0,0,0]]  
)  
+q*matrix(  
[[0,0,0,0,0,0,0],[0,0,0,0,0,0,0],[0,0,0,0,0,0,0],[0,0,0,0,0,0,0],  
[0,0,0,0,0,0,0],[0,0,0,0,0,1,0],[0,0,0,0,0,0,-1]]  
)
```

B

```

[1  x_0 x_1 x_2  x_3 x_4  0      ]
[x_0 x_2 x_3 x_5  x_6 x_7  0      ]
[x_1 x_3 x_4 x_6  x_7 x_8  0      ]
[x_2 x_5 x_6 x_9  x_10 x_11 0      ]
[x_3 x_6 x_7 x_10 x_11 x_12 0      ]
[x_4 x_7 x_8 x_11 x_12 x_13 0      ]
[0  0  0  0  0  0  1 - x_9 - x_13]

```

C=a*matrix.zero(7)

p.add_constraint(C<=B)

p.set_objective(a+8*b)

p.solver_parameter("show_progress", True)

opt = p.solve()

	pcost	dcost	gap	pres	dres	k/t
0:	0.0000e+00	-1.2062e+01	5e+01	2e+00	1e+00	1e+00
1:	-7.3207e+00	-8.8855e+00	7e+00	3e-01	2e-01	5e-01
2:	-7.6107e+00	-8.0355e+00	2e+00	9e-02	6e-02	2e-01
3:	-8.3282e+00	-8.4559e+00	5e-01	2e-02	2e-02	4e-02
4:	-8.3660e+00	-8.3732e+00	3e-02	1e-03	9e-04	2e-03
5:	-8.3694e+00	-8.3709e+00	6e-03	3e-04	2e-04	4e-04
6:	-8.3715e+00	-8.3721e+00	2e-03	1e-04	7e-05	2e-04
7:	-8.3719e+00	-8.3721e+00	4e-04	2e-05	2e-05	3e-05
8:	-8.3721e+00	-8.3721e+00	9e-05	4e-06	3e-06	7e-06
9:	-8.3721e+00	-8.3721e+00	6e-06	3e-07	2e-07	5e-07
10:	-8.3721e+00	-8.3721e+00	9e-07	5e-08	3e-08	7e-08

Optimal solution found.

opt

8.372143780723578

This is exactly the optimum found "by hand" above!