

## CVX demo

Sept 23 2019, AIMS S Africa short-course on optimization. Prof. Becker. This demonstrates basic usage of cvx (<http://cvxr.org/cvx/>) and how it obviates the need to re-write problems in standard form.

## Make some sample problems

```
n = 10;
m = 5;
rng(0); % set the seed
A = randn(m,n);
b = randn(m,1);
```

Solve

$$\min_x \|x\|_1 \quad \text{s.t.} \quad Ax = b$$

This is a linear program, but it is not in standard form, so standard solvers cannot solve it without rewriting it. Let us rewrite it (by hand):

$$\min_{s,x} \sum_i s_i \quad \text{s.t.} \quad Ax = b, x_i \leq s_i, -x_i \leq s_i$$

and to get this into standard form, we have still more work. Want it like:

$$\min_z c^T z \quad \text{s.t.} \quad Bz \leq f, Dz = d$$

where

$$z = [x, s]$$

Doing this in CVX is:

```
c = [zeros(n,1); ones(n,1)];
B = [ eye(n), -eye(n);
      -eye(n), -eye(n) ];
f = zeros(2*n,1);
D = [ A, zeros(m,n) ];
d = b;

cvx_quiet true % optional
cvx_precision best % optional
cvx_begin
    variable z(2*n)
    minimize dot(c,z)
    subject to
        B*z <= f
        D*z == b
cvx_end
```

and look at solution

```
x = z(1:n);  
x_slowAndPainful = x
```

```
x_slowAndPainful =  
    0.0000  
    0.3023  
   -0.0000  
   -0.9867  
   -0.1822  
   -0.0204  
   -0.0000  
   -0.3664  
    0.0000  
    0.0000
```

Try using some of the power of CVX

```
cvx_begin  
    variables x(n) s(n)  
    minimize sum(s)  
    subject to  
        abs(x) <= s  
        A*x == b  
cvx_end  
x_lessPainful = x;  
norm( x_lessPainful - x_slowAndPainful )
```

```
ans = 3.4811e-12
```

Now try with the full power of CVX

```
cvx_begin  
    variable x(n)  
    minimize norm(x,1)  
    subject to  
        A*x==b  
cvx_end  
  
norm( x - x_slowAndPainful )
```

```
ans = 3.4811e-12
```