Lab #1

1. Install and play with CVX[1]. Be sure to learn how to solve a least-squares problem.

2. *Reformulating constaints in* `cvx`. Each of the following `cvx` code fragments describes a convex constraint on the scalar variables `x`, `y`, and `z`, but violates the `cvx` rule set, and so is invalid. Briefly explain why each fragment is invalid. Then, rewrite each one in an equivalent form that conforms to the `cvx` rule set. In your reformulations, you can use linear equality and inequality constraints, and inequalities constructed using `cvx` functions. You can also introduce additional variables, or use LMIs. Be sure to explain (briefly) why your reformulation is equivalent to the original constraint, if it is not obvious.

   Check your reformulations by creating a small problem that includes these constraints, and solving it using `cvx`. Your test problem doesn't have to be feasible; it's enough to verify that `cvx` processes your constraints without error.

   Remark. This *looks* like a problem about 'how to use `cvx` software', or 'tricks for using `cvx`'. But it really checks whether you understand the various composition rules, convex analysis, and constraint reformulation rules.

   (a) `norm( [ x + 2*y , x - y ] ) == 0`

   (b) `square( square( x + y ) ) <= x - y`

   (c) `1/x + 1/y <= 1; x >= 0; y >= 0`

   (d) `norm([ max( x , 1 ) , max( y , 2 ) ]) <= 3*x + y`

   (e) `x*y >= 1; x >= 0; y >= 0`

   (f) `( x + y )^2 / sqrt( y ) <= x - y + 5`

   (g) `x^3 + y^3 <= 1; x>=0; y>=0`

   (h) `x+z <= 1+sqrt(x*y-z^2); x>=0; y>=0`

[1] At `http://www.stanford.edu/~boyd/cvx/`. If you do not have access to matlab, check out the Python version, CVXOPT.