MATH 5750/6880 OPTIMIZATION
HOMEWORK 3

1. **Interior point LP solver**

   Implement the primal-dual path following interior point algorithm for linear programs (Framework 14.1 p396 in Nocedal and Wright) in standard form:

$$\begin{aligned} &\min && c^T x \\ &\text{s.t.} && Ax = b \\ & && x \geq 0, \end{aligned} \tag{1}$$

   where $x, c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. Here are some tips:

   - You may use the interface: `function [x,y,z,flag] = lpipm(A,b,c,x0,y0,z0)`
   - Choice of the step length α: it is probably easier to use a fraction ρ of the smallest α of the two appearing in (14.36). That is use the same step size for the primal and dual variables and take for example $\rho = 0.9$.
   - Take $\sigma = 1/2$.
   - Test your code with a small example first (see Problem 2.), so forming the system matrix in (14.10) in Matlab is not too expensive. Then later when you know your code is working, you can get a significant speed improvement (that you will need for the application in Problem 3.) by solving the linear systems as in pp411-412 (14.44).
   - Print at each iteration: iteration number, duality gap, and the chosen step size α.
   - The linear systems to solve in (14.10) or (14.44) can get ill-conditioned if for example the slacks z become too small with respect to the variables x. This will happen in the application in Problem 3. (you will see warnings about a matrix being ill-conditioned), and you should stop the iterations when it happens. Otherwise you can get spurious solutions to the linear systems: the higher the condition number the less precision digits you can expect in a numerical solution.

     To see if A is ill-conditioned when solving the system $Ax = b$: instead of using Matlab's backslash (`x = A \ b;`), use `[x,icond] = linsolve(A,b);`. The output `icond` is the reciprocal of the condition number. If `icond < 1e-14`, stop the iterations of your interior point solver and return the current iterate.

2. **Test interior point LP solver**

   Check your interior point LP solver on a small LP in standard form with

$$A = \begin{bmatrix} 1 & -1 & 1 & 0 \\ 1 & -3 & 0 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 3 \\ 1 \end{bmatrix}, \quad c = [3, 1, 0, 0]^T$$

   (a) Find a feasible point (x_0, y_0, z_0) for this LP, s.t. $x_0 > 0$ and $z_0 > 0$.

   (b) Try your interior point solver with the above feasible point and also with $x_0 = [1, 1, 1, 1]^T$, $y_0 = [1, 1]^T$, $z_0 = [1, 1, 1, 1]^T$. Please include the convergence history.

   (c) Compare your solution to that of the simplex method implementation posted in the class website and Matlab's `linprog` (if you have the optimization toolbox).

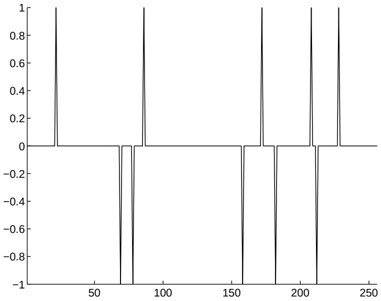

FIGURE 1. Sample sparse signal of length $n = 256$ and with $t = 10$ non-zero elements.

3. **Compressed Sensing application** (following `http://www.acm.caltech.edu/l1magic/`).

Let $x_{true} \in \mathbb{R}^n$ be a sparse signal (meaning it can represented with a few t terms of some basis). An example of a sparse signal is one that is mostly zeros except for $t \ll n$ non-zero entries which are ±1, see Figure 1 for an example. To generate one such signal in Matlab:

```
% random +/- 1 signal
xtrue = zeros(n,1);
q = randperm(n);
xtrue(q(1:t)) = sign(randn(t,1));
```

We would like to recover x_{true} from measurements $y \in \mathbb{R}^m$ which are obtained by doing $y = Ax_{true}$, where A is a given $m \times n$ matrix. Under certain assumptions on A it is possible to recover x_{true} *exactly* with overwhelming probability by solving the ℓ_1 minimization problem:

$$\begin{aligned} \min \quad & \|x\|_1 \\ \text{s.t.} \quad & Ax = y \\ & x \in \mathbb{R}^n \end{aligned} \tag{2}$$

where $\|x\|_1 = \sum_{i=1}^n |x_i|$. This problem can be recast as an LP by introducing the variables $z \in \mathbb{R}^n$, $z \geq 0$:

$$\begin{aligned} \min \quad & \sum_{i=1}^n z_i \\ \text{s.t} \quad & Ax = y \\ & -z \leq x \leq z \end{aligned} \tag{3}$$

(a) Transform the LP (3) to standard form (you will need to add slack variables and take into account that $x \in \mathbb{R}^n$ in general)

(b) Use your interior point solver to estimate x given some measurements y by solving the standard form of (3). Do this for the signal of ±1 generated as above and with the dimensions: $n = 128$, $m = 32$, $t = 5$ and $n = 256$, $m = 64$, $t = 10$. Compare the reconstructed x with x_{true} (they should be almost identical). For the measurement matrix A take a random orthonormal matrix which can be generated by taking the QR factorization of a matrix with iid Gaussian entries:

```
% measurement matrix: a random orthonormal matrix
A = randn(m,n);
[Q,R] = qr(A',0);
A = Q';
```