Some of the question below have instructions in **bold** that require you to make plots or answer questions. Create a short pdf document called "hmwk4.pdf" with your answers for each question. You don't need to make an elaborate writeup, just put your plots into the pdf with labels indicating what they are, and answer questions when prompted. **Put your name at the top of your writeup!**

1. Write an "unwrapped ADMM" solver for the support-vector machine problem

$$\text{minimize} \quad \frac{1}{2}\|x\|^2 + Ch(Ax)$$

where $h(z) = \sum_i \max\{1 - z_i, 0\}$ is the hinge loss function, and $A = LD$ is the product of the (diagonal) label matrix with the data matrix. If the ith training vector is d_i and has binary (± 1) label l_i, then the ith row of A is $l_i d_i$.

Your solver should use the change of variables $y \leftarrow Ax$. Then find a saddle point of the (scaled) augmented Lagrangian

$$L_\tau(x, y, \lambda) = \frac{1}{2}\|x\|^2 + Ch(y) + \frac{\tau}{2}\|Ax - y + \lambda\|^2$$

where λ is a vector of Lagrange multipliers. Proceed by the following steps.

- Minimize for x:

$$x^{k+1} = \arg\min_x \frac{1}{2}\|x\|^2 + \frac{\tau}{2}\|Ax - y + \lambda\|^2.$$

This requires solution of a linear system (a least-squares problem with a ridge penalty). You should cache the factorization so that it is only computed once.

- Minimize for y:

$$y^{k+1} = \arg\min_y Ch(y) + \frac{\tau}{2}\|Ax - y + \lambda\|^2.$$

This can be done easily if you know the proximal operator of $h(\cdot)$ which is given by

$$\text{prox}_h(z, \delta) = \arg\min_u h(u) + \frac{1}{2\delta}\|u - z\|^2.$$

The solution can be written entry-wise as

$$\text{prox}_h(z, \delta)_k = z_k + \max\{\min\{1 - z_k, \delta\}\}.$$

- Update λ :

$$\lambda^{k+1} = \lambda^k + (Ax^{k+1} - y^{k+1}).$$

Use your method to solve a classification problem with 100 features, and N vectors (you can pick N). **Create a convergence curve that plots the residuals vs iteration count. The primal and dual residuals for this problem are given by**

$$p^k = \|Ax^k - y^k\|$$
$$d^k = \|\tau A^T(y^k - y^{k-1})\|.$$

Automatically stop your iteration when residuals get "small." One possible stopping rule is to stop when $p^k \le 10^{-3} \max_{i<k} p^i$, and $d^k \le 10^{-3} \max_{i<k} d^i$. However you may also just stop when $p^k, d^k < \epsilon$ for some small ϵ.

Choose a reasonably value for τ that gives good performance. **How big can you make N and still have the solver terminate within 30 seconds?**

2. Create a blur stencil that is simply a 5x5 square of ones. Create a test image (or just download Lena from google images). Convolve your image with the stencil to get a blurred image. Add a small amount of noise to the image (about 1% of the max signal should do fine).

   Now, deblurr your image by solving the total variation problem

$$\underset{x}{\text{minimize}} \quad \mu|\nabla x| + \|Kx - f\|^2$$

   where K is a blur operation and f is the blurred image. Your solver should use ADMM (aka Split Bregman). All sub-problems should be solved in closed form using shrinkage or fast Fourier transforms. If you need a reference, see the experiments section of the paper "Fast Alternating Direction Optimization Methods."

   In your writeup, show the original, blurred, and deblurred images. Also, plot convergence curves for ADMM.