Assignment 5-Coding Part: Due Friday 3/23

I) Complete the `sorSolve.m` function that implements the Succesive Over-Relaxation (SOR) method for iterative solution of $Ax = b$.

Key Idea of SOR method: We have a constant[1] $0 < \omega < 2$. At each iteration of SOR, we compute:

$$x_{new} = (\omega D + L)^{-1} (\omega b - [\omega U + (\omega - 1)D]x_{old})$$

Then we copy over x_{old} with x_{new}.

A practical implementation is:

1) Extract L, D, U from A

2) Store quantities used repeatedly:
$v = \omega b$
$R = \omega U + (\omega - 1)D$
$C = \omega D + L$ (C will be lower triangular!)

3) Let $x = x_0$ (where x_0 is user supplied initial guess)

4) Loop: For $k = 1 : maxIters$
Let $t = v - Rx$ (temp. vector)
Solve $Cx = t$ by forward subsitution.
Calculate Residual $r = b - Ax$
Break the loop if $\|r\|_\infty$ is small enough

Suggestion: Copy the contents of jacobiSolve.m into sorSolve.m; implement the changes listed in the comments

II) Complete the `threshold.m` function. See comments in that function for details.

III) Complete the `saveImg.m` function. See comments in that function for details.

What to turn in: The 3 files `sorSolver.m`, `threshold.m`, `saveImg.m` AND screenshots figures of denoised and thresholded images (figures 2 and 3) after executing the following at the command line:

```
>> denoiseImg('hwImg.png',0.006);
```

[1] $\omega = 1$ is the Gauss-Seidel method.