

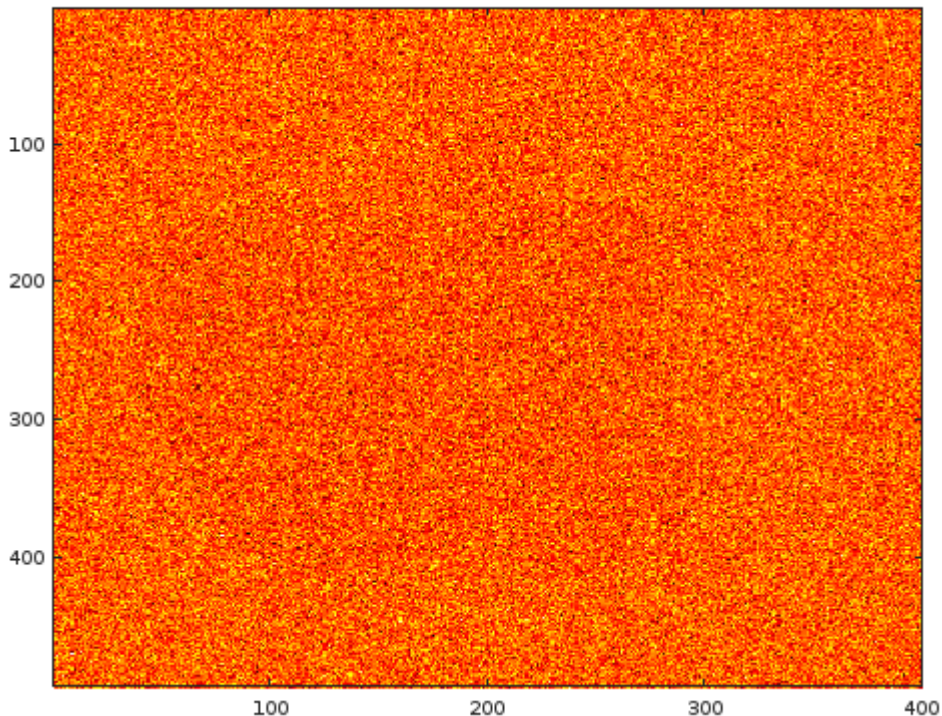
If you wrote the **sorSolver** and **threshold** functions correctly, then you get the following output for the image **testImg.png**

0) Function Call + Printed Output

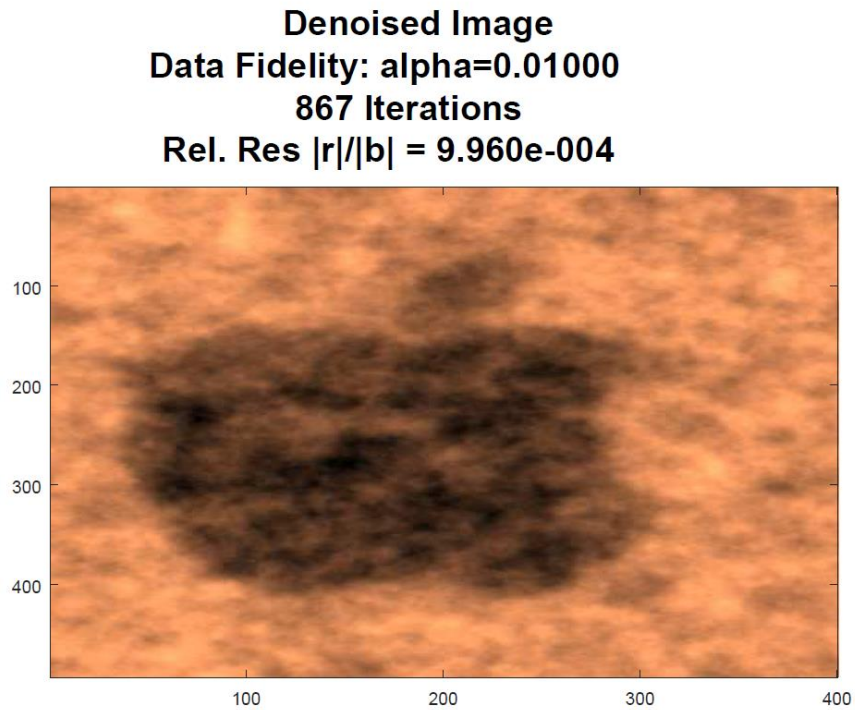
```
Command Window
>> denoiseImg('testImg.png',0.01);
Image Loaded
Filter Matrix Construction Complete!
Starting Solver...
Iter 50: Rel. Residual |r|/|b| = 8.871e-001
Iter 100: Rel. Residual |r|/|b| = 5.844e-001
Iter 150: Rel. Residual |r|/|b| = 3.856e-001
Iter 200: Rel. Residual |r|/|b| = 2.545e-001
Iter 250: Rel. Residual |r|/|b| = 1.680e-001
Iter 300: Rel. Residual |r|/|b| = 1.109e-001
Iter 350: Rel. Residual |r|/|b| = 7.319e-002
Iter 400: Rel. Residual |r|/|b| = 4.831e-002
Iter 450: Rel. Residual |r|/|b| = 3.188e-002
Iter 500: Rel. Residual |r|/|b| = 2.104e-002
Iter 550: Rel. Residual |r|/|b| = 1.389e-002
Iter 600: Rel. Residual |r|/|b| = 9.165e-003
Iter 650: Rel. Residual |r|/|b| = 6.048e-003
Iter 700: Rel. Residual |r|/|b| = 3.991e-003
Iter 750: Rel. Residual |r|/|b| = 2.634e-003
Iter 800: Rel. Residual |r|/|b| = 1.738e-003
Iter 850: Rel. Residual |r|/|b| = 1.147e-003
Terminating at iteration 867 due to residual tolerance being met
>>
```

1) The noisy image looks awful

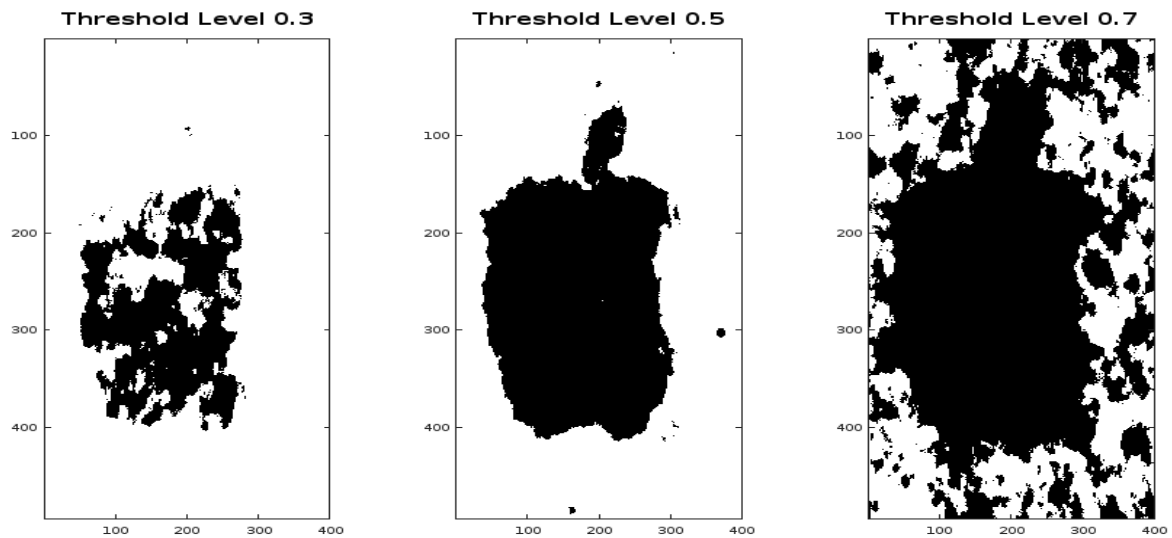
Noisy Image



2) The denoised image (Figure 2) should be the following



3) If your threshold function is working, the thresholded images should be:



A smaller value for the data fidelity parameter should produce a smoother image, but at more computational expense.

Denoised Image
Data Fidelity: $\alpha=0.00200$
4321 Iterations
Rel. Res $|r|/|b| = 9.994\text{e-}004$

