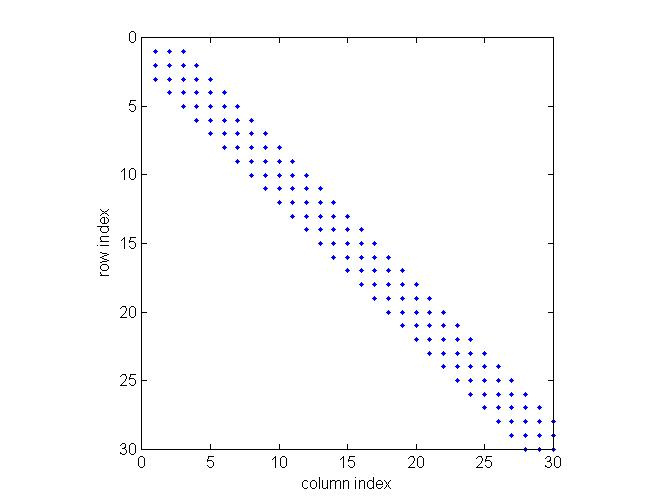


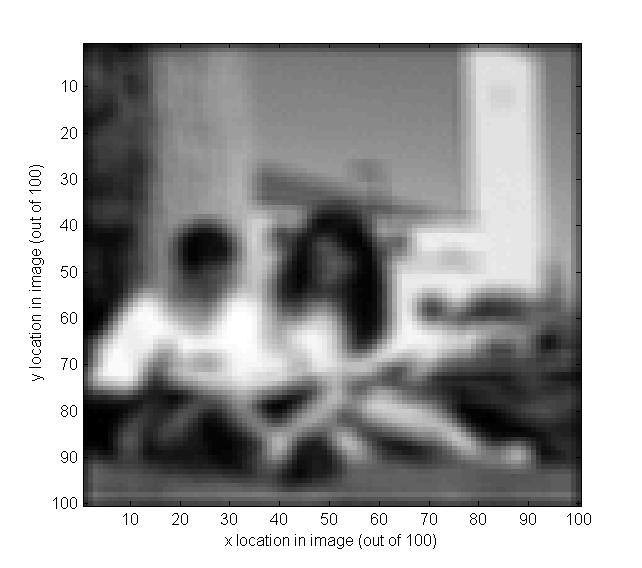
**1a-1 (full structure)**



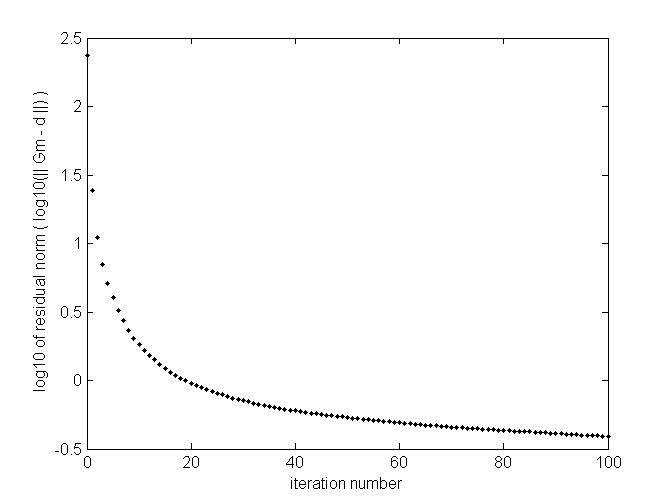
**1a-2 (zoomed in)**

**1a-3 (zoomed in more)**

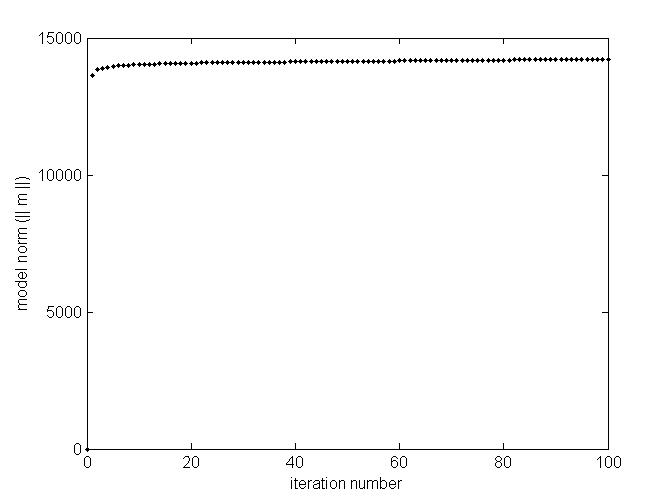
**Figure 1a**. Sparsity structure of the image blurring matrix (G) for problem 1. Note that as we zoom in toward the main diagonal in the upper-left corner of the matrix (**1a-2**), we see five distinct nonzero bands, and within these we find an even finer 5-banded structure (**1a-3**). This pattern would be typical of a 5 x 5 square stencil used in many finite difference routines (or here, for local blurring).



**Figure 1c**. Here is the original (blurred) image from problem 1 (vector **d**).



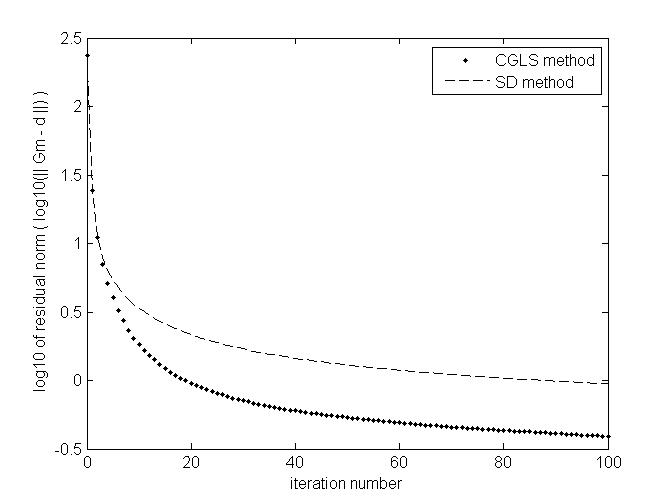
**Figure 1d-1**. The log(10) of residual error norm is plotted here vs. iteration number for the manually-coded CGLS routine in Problem 1 (Part D).



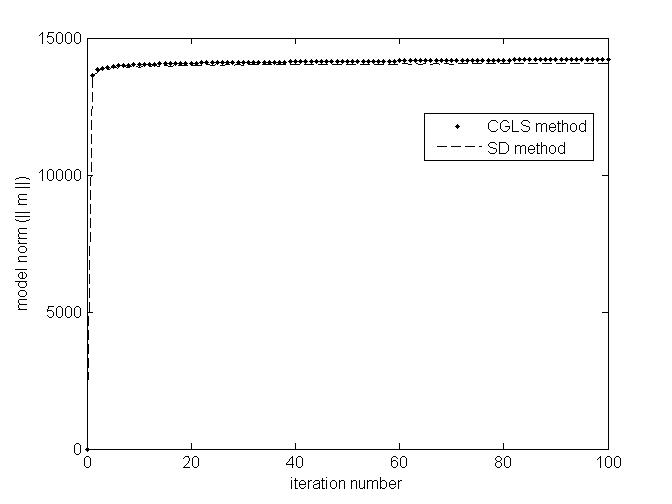
**Figure 1d-2**. Model norm is plotted here vs. iteration number for Part D of Problem 1 (this corresponds with plot **1d-1** above).

|  |  |
| --- | --- |
| **C:\Users\Brad Martin\Desktop\CU Boulder\Spring 2014\Inverse Methods\HW4\fig1e1.jpg**  **model image after 10 CGLS iterations** | **C:\Users\Brad Martin\Desktop\CU Boulder\Spring 2014\Inverse Methods\HW4\fig1e2.jpg**  **image after 25 CGLS iterations** |
| **C:\Users\Brad Martin\Desktop\CU Boulder\Spring 2014\Inverse Methods\HW4\fig1e3.jpg**  **image after 40 CGLS iterations** | **C:\Users\Brad Martin\Desktop\CU Boulder\Spring 2014\Inverse Methods\HW4\fig1e4.jpg**  **image after 55 CGLS iterations** |

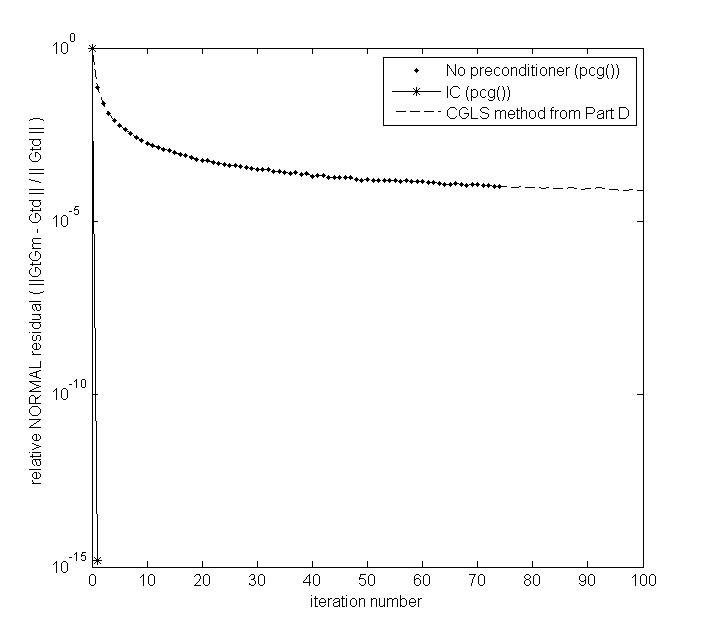
**Figure 1e**. The pictures above represent evolution of the model image in Problem 1 after 10, 25, 40, and 55 CGLS algorithm iterations. As seen here, image sharpness increases with iteration number, but noise increases as well (the image becomes more grainy). The solution after 25 iterations seems to offer a good compromise between sharpness and noise.



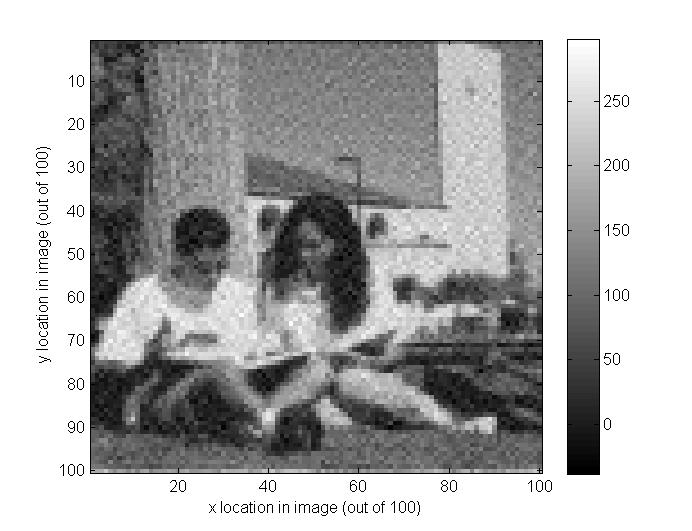
**Figure 1f-1**. The log of residual error norm is plotted here vs. iteration number for the manually-coded CGLS routine in Problem 1 (Part D) and for its adaptation into a less efficient SD algorithm (Part F).



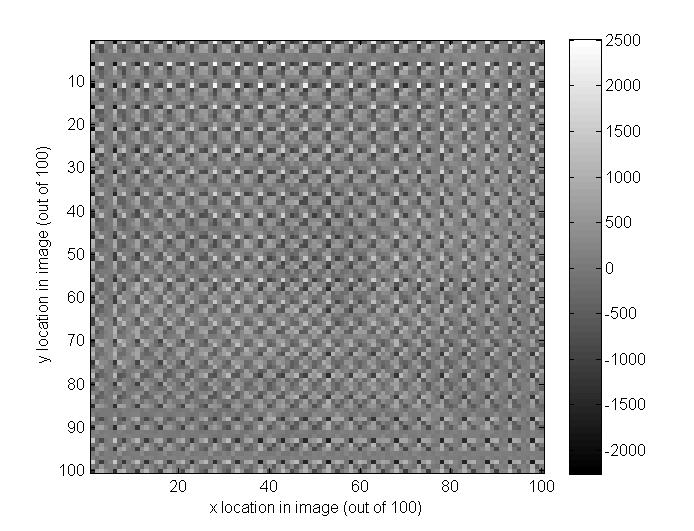
**Figure 1f-2**. Model norm is plotted here vs. iteration number for Part D (CGLS) and Part F (SD) of Problem 1 (this corresponds with plot **1f-1** above).



**Figure 1h-1**. Relative residual norm is plotted above vs. iteration number for 1) the MATLAB pcg() function applied to Problem 1, using no preconditioner; 2) the MATLAB pcg() function, using an incomplete Cholesky preconditioner; and 3) data from the manually-programmed CGLS routine in Part D (for reference).



**Figure 1h-2**. This is the model image returned from MATLAB’s pcg() routine after 100 iterations without a preconditioner. It’s a bit noisy, but that’s what we’d expect given our results in Part D (manually coded CGLS).



**Figure 1h-3**. This is the model image returned from MATLAB’s pcg() routine after just 1 iteration when using an incomplete Cholesky preconditioner. Although the residual norm of this solution is quite small (on the order of 1E-14), the solution is dominated by noise (as we’ve seen in a lot of non-truncated solutions in class).