Terms and conditions

Previously you have seen how to blur an image, represented as an $m \times n$ pixel intensity matrix $\boldsymbol{X}$, through multiplying by a matrix on each side:

$$\boldsymbol{Z} = \boldsymbol{V}\boldsymbol{X}\boldsymbol{H}, \tag{1}$$

where $\boldsymbol{V} = (\boldsymbol{B}_m)^k$, $\boldsymbol{H} = (\boldsymbol{B}_n)^k$, B_i is the $i \times i$ matrix made by your function `blurmatrix`, and k is a positive integer. You have also seen that deblurring can be accomplished by multiplying by matrix inverses on each side of $\boldsymbol{Z}$ (as computed equivalently by solving linear systems). However, while the restoration seems perfect for $k = 1$, it fails completely for larger k.

Matrix condition numbers explain these observations. The blurred matrix $\boldsymbol{Z}$ is perturbed by an amount comparable to machine precision. For $k = 1$, the condition numbers of $\boldsymbol{V}$ and $\boldsymbol{H}$ are not large enough to amplify this error up to the same order of magnitude as $\boldsymbol{X}$ itself, so the noise is not perceived. But at some $k > 1$, the condition numbers are so large that the noise is amplified enough to overwhelm the expected result.

Let $\boldsymbol{x}$ be a single column of $\boldsymbol{X}$. For the case of vertical blurring only, we have $\boldsymbol{z} = \boldsymbol{V}\boldsymbol{x}$ as a column of $\boldsymbol{Z}$. We solve $\boldsymbol{V}\boldsymbol{y} = \boldsymbol{z}$ for $\boldsymbol{y}$, which is mathematically the same as $\boldsymbol{x}$. Due to machine precision, though, the perturbation to $\boldsymbol{z}$ causes an error satisfying

$$\frac{\|\boldsymbol{y} - \boldsymbol{x}\|}{\|\boldsymbol{x}\|} \le \kappa(\boldsymbol{V})\varepsilon_{\text{mach}}, \tag{2}$$

with κ being the matrix condition number.

Preparation

Read Section 2.8. Make sure your working `blurmatrix.m` is available.

Goals

You will compute condition numbers of blur matrix powers and compare them to the errors of repeated blur/deblur operations.

Procedure

1. The condition number of a matrix has two factors, $\|\boldsymbol{A}\|$ and $\|\boldsymbol{A}^{-1}\|$. First you will show that the $\|\boldsymbol{A}\|$ term makes no trouble. For $n = 50, 100, 150, \ldots, 800$, plot $\|\boldsymbol{B}_n\|$ versus n.

2. For the same n as in step 1, plot $\kappa(\boldsymbol{B}_n)$ versus n. You should use a log-log scale for this graph and get essentially a straight line. This implies that $\log\kappa \approx a\log n + b$, or $\kappa \approx Cn^p$.

3. Let $\boldsymbol{V} = \boldsymbol{B}_{100}$. For $k = 1, 2, \ldots, 8$, plot $\kappa(\boldsymbol{V}^k)$ as a function of k. This time the graph is straight on a semi-log scale, which implies $\kappa \approx Cq^k$.

4. Let $\boldsymbol{x}$ be a random vector of length 100. For $k = 1, 2, \ldots, 8$, let $\boldsymbol{z} = \boldsymbol{V}\boldsymbol{x}$ and then solve $\boldsymbol{V}\boldsymbol{y} = \boldsymbol{z}$ for $\boldsymbol{y}$. Record the relative error in the result. Then make a table showing both sides of the inequality (2).