CAAM 453/550: Numerical Analysis I - Fall 2021
Homework 6 - due by 5pm on Wednesday, October 6, 2021

Instructions: You may use any of the code on the Canvas page. Turn in all modified/new MATLAB/python code (scripts and functions) and all output generated by your code. All code must be commented, and it must be clear what your output is and why you are submitting it. Additionally, all plots must be labeled.

For any problems that do not require coding, either turn in handwritten work or typeset work using LaTeX or some other typsetting software. Please do not turn in math as commented MATLAB code or math that has been typed in a word processor!

MATLAB code fragments are provided in some problems. If you program in python, you may replace them corresponding numpy or scipy code.

CAAM 453 students are to complete problems 1 - 3. (70 points)
CAAM 550 students are to complete problems 1 - 4. (90 points)

CAAM 453 may complete additional problems "for fun," but you will not receive additional credit.

Problem 1 (30 points) Let $A \in \mathbb{R}^{n\times n}$ be a square, nonsingular matrix with SVD $A = U\Sigma V^T$ and singular values $\sigma_1 \geq \ldots \geq \sigma_n > 0$.

(a) (5 points) Describe how to use the SVD of A to solve a linear system $Ax = b$.

(b) (5 points) Let $\delta \in \mathbb{R}^n$ be a perturbation of the right hand side and let $\widetilde{x} \in \mathbb{R}^n$ solve $A\widetilde{x} = b + \delta$. Express the solution error $\varepsilon = \widetilde{x} - x \in \mathbb{R}^n$ in terms of the SVD of A and of the right hand side perturbation δ.

(c) (10 points) Use your expression in (b) to show that

$$\|\varepsilon\|_2 \leq \frac{1}{\sigma_n}\|\delta\|_2.$$

(d) (10 points) Consider the linear system $Ax = b$, with

$$A = \begin{pmatrix} 1 & t_1 & t_1^2 & \cdots & t_1^{n-1} \\ 1 & t_2 & t_2^2 & \cdots & t_2^{n-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & t_n & t_n^2 & \cdots & t_n^{n-1} \end{pmatrix} \in \mathbb{R}^{n\times n} \quad \text{and} \quad t_i = -1 + 2\frac{i-1}{n-1}, \quad i = 1,\ldots,n. \tag{1}$$

The matrix (1) is called a *Vandermonde matrix* and it arises, e.g., in polynomial interpolation and other applications. We construct the right hand side b so that the exact solution is $x = (1,1,\ldots,1)^T$, i.e.,

$$b_i = \sum_{j=1}^{n} a_{ij}\cdot 1 = \sum_{j=1}^{n} t_i^{j-1} = \begin{cases} \frac{1-t_i^n}{1-t_i} & \text{if } t_i \neq 1, \\ n & \text{if } t_i = 1. \end{cases}$$

For $n = 10,\ldots,30$, compute the solutions x_{comp} of $Ax = b$ using, e.g., Matlab's `A \ b`, and generate a table that shows n, $1/\sigma_n$, $\|x - x_{\text{comp}}\|_2$, and $\|x - x_{\text{comp}}\|_2\sigma_n$.

The numerical solution of a linear system is subject to floating point arithmetic errors. So called backward analysis allows us to interpret the computed (floating point arithmetic) result x_{comp} as the exact solution of a linear system $Ax_{\text{comp}} = b + \delta$, where the size $\|\delta\|_2$ is proportional to the floating point arithmetic error (which is $\approx 10^{-14}$ in double precision floating point arithmetic, which is what Matlab and numpy (default) use).

Do your computational results match the bound you have derived in (c)?

Problem 2 (10 points) Download a few flags from `https://www.cia.gov/the-world-factbook/references/flags-of-the-world` (accessed September 27, 2021) and compute low rank representations. What is a 'good' rank for the flags you have chosen?

Problem 3 (30 points) The set-up for this problem is identical to that of Problem 3 in Homework 2, but is repeated here for completeness. You may reuse code from Problem 3 in Homework 2.

We want to reconstruct an image from a blurred version using regularized least squares. The true image is represented by a function $f : [0,1] \to [0,1]$ (think of $f(\xi)$ as the gray scale of a one-dimensional image at ξ with gray values scaled to $[0,1]$). The blurred image $g : [0,1] \to \mathbb{R}$ is given by

$$\int_0^1 k(\xi_1, \xi_2) f(\xi_2)\, d\xi_2 = g(\xi_1), \quad \xi_1 \in [0,1], \tag{2}$$

where $k : [0,1]^2 \to [0,\infty)$ is given by

$$k(\xi_1, \xi_2) = \frac{1}{\gamma\sqrt{2\pi}} \exp\big(-(\xi_1 - \xi_2)^2/(2\gamma^2)\big), \tag{3}$$

with $\gamma > 0$. The map $f \mapsto g = \int_0^1 k(\cdot, \xi_2) f(\xi_2) d\xi_2$ is called a convolution and k is called the (convolution) kernel.

Given the kernel k and the blurred image g we want to find f. The smaller the parameters $\gamma > 0$ in the kernel (3) the less blurred the image is. See Figure 1.

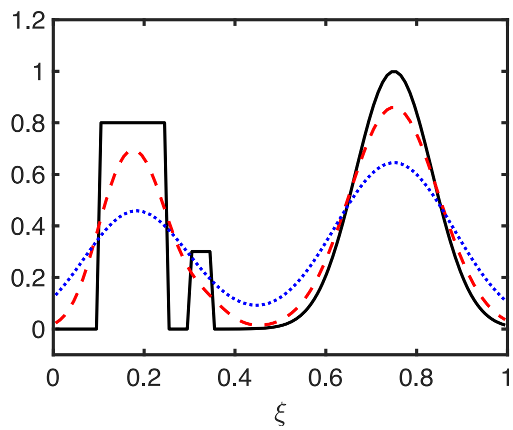

Figure 1: The image f (black solid line) and the blurred images g computed from (2) with $\gamma = 0.05$ (red dashed line) and $\gamma = 0.1$ (blue dotted line)

In the following computations we use

$$\gamma = 0.05.$$

To discretize the problem, we divide $[0,1]$ into n equidistant intervals of length $h = 1/n$. Let $\xi_i = (i - \frac{1}{2})h$ be the midpoint of the ith interval. We approximate f and g by piecewise constant functions,

$$f(\xi) \approx \sum_{i=1}^{n} f_i \chi_{[(i-1)h, ih]}(\xi), \qquad g(\xi) \approx \sum_{i=1}^{n} g_i \chi_{[(i-1)h, ih]}(\xi),$$

where χ_I is the indicator function on the interval I. We insert these approximations into (2) and approximate the integral by the midpoint rule. This leads to the $n \times n$ linear system

$$\mathbf{K}\mathbf{f} = \mathbf{g}, \tag{4}$$

where

$$\mathbf{f} = (f_1, \ldots, f_n)^T, \quad \mathbf{g} = (g_1, \ldots, g_n)^T,$$

and

$$\mathbf{K}_{ij} = h k(\xi_i, \xi_j), \quad i, j = 1, \ldots, n.$$

Let

$$n = 100,$$

construct the true image $\mathbf{f}^{\text{true}}$, the resulting blurred image $\mathbf{g}^{\text{true}} = \mathbf{K}\mathbf{f}^{\text{true}}$, and blurred image $\mathbf{g}$ with 0.1% noise using

```
ftrue       = zeros(n,1);
ftrue       = exp( -(xi -0.75).^2 * 70 );
ind         = (0.1<=xi) & (xi<=0.25);
ftrue(ind) = 0.8;
ind         = (0.3<=xi) & (xi<=0.35);
ftrue(ind) = 0.3;
% blurred image
gtrue       = K*ftrue;
% data perturbation of 0.1%
rng('default')
gerr    = 0.001*(0.5-rand(size(gtrue))).*gtrue;
g       = gtrue + gerr;
```

The true and blurred image (with noise) are shown in the left plot in Figure 2. We want to recover $\mathbf{f}^{\text{true}}$ from $\mathbf{g}$ and (4). Solving the linear system (4) using MATLAB 's backslash leads to a highly oscillatory function, indicated by the blue dashed lines in the right plot in Figure 2.

The image recovery leads to the linear least squares problem

$$\min_{\mathbf{f} \in \mathbb{R}^n} \frac{1}{2} \|\mathbf{K}\mathbf{f} - \mathbf{g}\|_2^2,$$

where $\mathbf{g} = \mathbf{g}^{\text{true}} + \mathbf{g}^{\text{err}}$.

i. (10 points) Compute the singular value decomposition $\mathbf{K} = \mathbf{U}\Sigma\mathbf{V}^T$. The columns of $\mathbf{U}$ and $\mathbf{V}$ are denoted by $\mathbf{u}_1, \ldots, \mathbf{u}_n$ and $\mathbf{v}_1, \ldots, \mathbf{v}_n$. The singular values are $\sigma_1 \geq \ldots \geq \sigma_n \geq 0$.

   Plot the singular values σ_j of $\mathbf{K}$, $j = 1, \ldots, n$, and $\mathbf{u}_j^T \mathbf{g}^{\text{true}}$, $\mathbf{u}_j^T \mathbf{g}^{\text{err}}$, $j = 1, \ldots, n$ (`semilogy` plot).

   Plot the right singular vectors $v_1, v_{15}, v_{30}, v_{45}, v_{60}, v_{75}$. Hint: In MATLAB use `subplot(2,3,1)` etc. You may need to resize your figure using something like the following (depending on your screen, you may need to change the position).

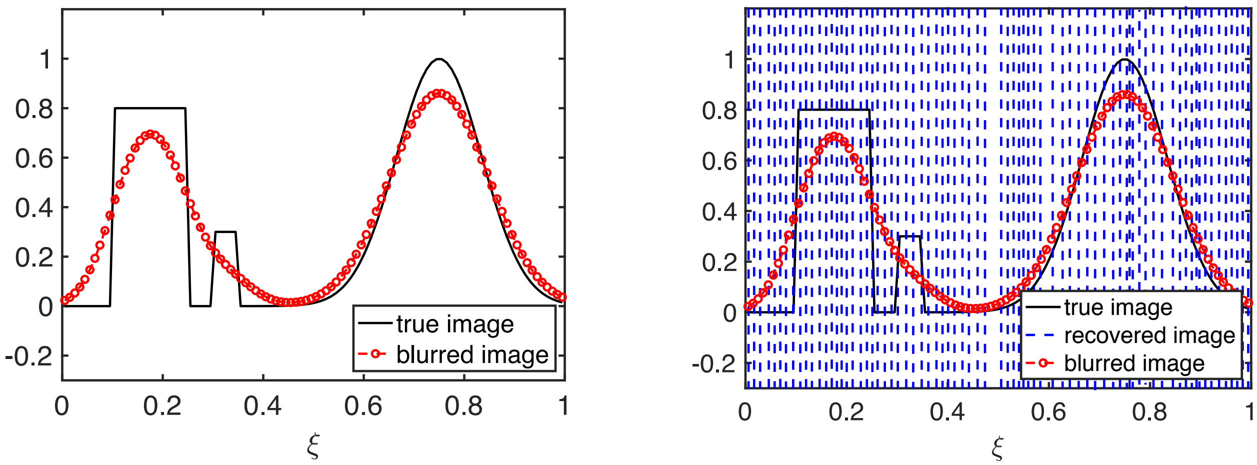

Figure 2: Left plot: The true image $\mathbf{f}^{\text{true}}$ and the blurred image $\mathbf{g}$. Right plot: The true image $\mathbf{f}^{\text{true}}$, the blurred image $\mathbf{g}$, and the recovered image $\mathbf{f} = \mathbf{K}^{-1}\mathbf{g}$ obtained by solving (4)

```
hFig = figure( 3 ); clf; set( hFig, 'Position',[100 800 1400 700] );
subplot(2,3,1); plot( xi, V(:,1)); text(0.5,-0.05,'v_1','FontSize',20);
subplot(2,3,2); plot( xi, V(:,15)); text(0.5,0.17,'v_{15}','FontSize',20);
```

Use the SVD to explain why the recovered image $\mathbf{f} = \mathbf{K}^{-1}\mathbf{g}$ Shown in Figure 2 is such a poor approximation of the true image.

ii. (10 points) Define

$$\mathbf{f}_k = \sum_{i=1}^{k} \frac{\mathbf{u}_i^T \mathbf{g}}{\sigma_i} \mathbf{v}_i$$

and plot $\|\mathbf{f}_k\|_2$, $\|\mathbf{K}\mathbf{f}_k - \mathbf{g}\|_2$, $k = 1, \ldots, n$, as well as the constant $\|\mathbf{g} - \mathbf{g}^{\text{true}}\|_2$ in a `semilogy` plot.

Plot the recovered images corresponding to $\mathbf{f}_1, \mathbf{f}_{15}, \mathbf{f}_{30}, \mathbf{f}_{45}, \mathbf{f}_{60}, \mathbf{f}_{75}$ and the true image. Hint: In MATLAB use `subplot(2,3,1)` etc. The format of each plot should follow that of the right plot in Figure 2.

iii. (5 points) Find k_* such that

$$\Big| \|\mathbf{K}\mathbf{f}_{k_*} - \mathbf{g}\|_2 - \|\mathbf{g}^{\text{err}}\|_2 \Big| = \min_k \Big| \|\mathbf{K}\mathbf{f}_{k_*} - \mathbf{g}\|_2 - \|\mathbf{g}^{\text{err}}\|_2 \Big|.$$

What is k_*? Plot the true image, the image corresponding to $\mathbf{f}_{k_*}$ and the blurred image. (The format of your plot should follow that of the right plot in Figure 2.)

Morozov's Discrepancy principle, introduced in Problem 3 in Homework 2 would choose k_* according to this criterion. Is this a good choice in this case?

iv. (5 points) Plot $\|\mathbf{f}_k\|_2$ versus $\|\mathbf{K}\mathbf{f}_k - \mathbf{g}\|_2$ ($\|\mathbf{K}\mathbf{f}_k - \mathbf{g}\|_2$ on x-axis). The resulting curve should look like an 'L' and is referred to as the L-curve. The L-curve criterion picks k_* such that $(\|\mathbf{K}\mathbf{f}_{k_*} - \mathbf{g}\|_2, \|\mathbf{f}_{k_*}\|_2)$ is in the 'corner' of the L-curve.

What is k_* according to the L-curve criterion? (You may find this point manually, by evaluating $(\|\mathbf{K}\mathbf{f}_{k_*} - \mathbf{g}\|_2, \|\mathbf{f}_{k_*}\|_2)$ for several k_* and plotting the corresponding point on the L-curve.

Plot the true image, the image corresponding to $\mathbf{f}_{k_*}$ and the blurred image. The format of your plot should follow that of the right plot in Figure 2.

Problem 4 (20 points - CAAM 550 only) Let $A = U\Sigma V^T$ with

$$\Sigma = \begin{pmatrix} \sigma_1 & & & & & \\ & \ddots & & & & \\ & & \sigma_r & & & \\ & & & 0 & & \\ & & & & \ddots & \\ & & & & & 0 \end{pmatrix}.$$

be the singular value decomposition of the square matrix $A \in \mathbb{R}^{n\times n}$.

For $k \leq r$ we define

$$x_k = \sum_{i=1}^{k} \sigma_i^{-1} u_i^T b\, v_i.$$

(10 points) Show that

$$\|x_{k-1}\|_2 \leq \|x_k\|_2$$

(10 points) and

$$\|Ax_{k-1} - b\|_2 \geq \|Ax_k - b\|_2$$

for all $k = 2, \ldots, r$.