Review Problems for Final Exam — Math 3607

Tae Eun Kim Summer 2020

Below are problems from numerical analysis covering Gaussian elimination through the singular value decomposition; these are for the written part of the final exam. For the online part, please review all 7 quizzes.

Problem 1. (Gaussian Elimination by Hand)

Solve the following matrix equation by hand using partial pivoting:

$$\begin{bmatrix} 1 & -2 & 1 \\ 0 & 2 & 2 \\ -2 & 4 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 4 \\ 2 \end{bmatrix}.$$

Let

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \\ 16 & 17 & 18 & 19 & 20 \\ 21 & 22 & 23 & 24 & 25 \end{bmatrix}.$$

Denote the i^{th} row and the j^{th} column of A by $\mathcal{R}_i$ and $\mathcal{C}_j$ respectively.

(a) Multiply A by a permutation matrix P to interchange $\mathcal{R}_1$ with $\mathcal{R}_4$. Write out P explicitly.

(b) Multiply A by a permutation matrix P to interchange $\mathcal{C}_1$ with $\mathcal{C}_4$. Write out P explicitly.

(c) Multiply A by two permutation matrices P to interchange $\mathcal{C}_1$ with $\mathcal{C}_4$ and $\mathcal{R}_1$ with $\mathcal{R}_4$. Write out P explicitly.

(d) Write down MatLab statements for the previous parts, that is, create `A` and then permute its rows/columns as indicated WITHOUT using matrix multiplication.

(e) ($*$) Find a permutation matrix which moves $(\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3, \mathcal{R}_4)$ to $(\mathcal{R}_2, \mathcal{R}_3, \mathcal{R}_4, \mathcal{R}_1)$ respectively, leaving $\mathcal{R}_5$ unmoved. What is the smallest positive integer k such that $P^k = I$? Write this permutation as a product of $\boxed{\text{elementary}}$ permutation matrices.

Problem 3. (Backward Substitution)

Let $U \in \mathbb{R}^{n\times n}$ be an upper triangular matrix whose (i,j)-entry is denoted by $u_{i,j}$.

(a) Write a MatLab function `back_subs` which solves the matrix equation $U\mathbf{x} = \mathbf{y}$ using *backward substitution*. The function takes U and $\mathbf{y}$ as input arguments and produces $\mathbf{x}$ as an output argument.

(b) Show that the cost of solving $U\mathbf{x} = \mathbf{y}$ via backward substitution is approximately n^2 *flops* for large n.

Problem 4. (Properties of Gaussian Transformation Matrices)

Let $\{\mathbf{e}_j \in \mathbb{R}^n \mid j \in \mathbb{N}[1,n]\}$ be the standard unit basis of $\mathbb{R}^n$, i.e. $\mathbf{e}_1 = (1,0,0,\cdots,0)^{\mathrm{T}}$, $\mathbf{e}_2 = (0,1,0,\cdots,0)^{\mathrm{T}}$, $\ldots$, $\mathbf{e}_n = (0,0,0,\cdots,1)^{\mathrm{T}}$. Let $1 \leqslant j < i \leqslant n$. Show that the inverse of the elementary Gaussian transformation matrix of the form $G_j = I + a_{i,j}\mathbf{e}_i\mathbf{e}_j^{\mathrm{T}}$ is given by

$$G_j^{-1} = I - a_{i,j}\mathbf{e}_i\mathbf{e}_j^{\mathrm{T}}.$$

(*Hint:* You may find $\mathbf{e}_j^{\mathrm{T}}\mathbf{e}_i = \delta_{i,j}$ to be useful.)

Problem 5. (PLU Factorization by Hand)

Find the PLU-factorization of the matrix

$$A = \begin{bmatrix} 2 & -1 & 3 \\ 10 & -7 & 10 \\ -6 & 4 & -5 \end{bmatrix},$$

by Gaussian elimination with partial pivoting. That is, find matrices P (permutation matrix), L (unit lower triangular matrix), and U (upper triangular matrix) such that $PA = LU$. Do this by hand.

(*Hint:* You may use the results from the previous problem.)

Write a MatLab script carrying out the LDL factorization of the following symmetric matrix A:

$$A = \begin{bmatrix} 2 & 4 & 4 & 2 \\ 4 & 5 & 8 & -5 \\ 4 & 8 & 6 & 2 \\ 2 & -5 & 2 & -26 \end{bmatrix}.$$

That is, write a program that calculates a unit lower triangular matrix L and a diagonal matrix D satisfying $A = LDL^{\mathrm{T}}$. Assume that A is already stored in MatLab.

Let

$$A = \begin{bmatrix} 1 & 2 \\ 0 & 3 \end{bmatrix}.$$

(a) Calculate $\|A\|_1$, $\|A\|_2$, $\|A\|_\infty$, and $\|A\|_F$ all by hand.

(b) Find a vector $\mathbf{x}$ satisfying $\|\mathbf{x}\|_1 = 1$ and $\|A\mathbf{x}\|_1 = \|A\|_1$.

(c) Imagine that MatLab does not offer `norm` function and you are writing one for others to use, which begins with

```
function mat_norm(A, j)
% mat_norm   computes matrix norms
% Usage:
%    mat_norm(A, 1) returns the 1-norm of A
%    mat_norm(A, 2) is the same as mat_norm(A)
%    mat_norm(A, 'inf') returns the infinity-norm of A
%    mat_norm(A, 'fro') returns the Frobenius norm of A
```

Complete the program. (*Hint:* To handle the second input argument properly which can be a number or a character, use `ischaracter` and `strcmp`.)

Problem 8. (Lagrange and Newton Polynomial Interpolation (1))

A set of data points given in the table below is to be interpolated by a polynomial:

x_j	2	3	5	8
y_j	3	-2	12	3

(a) Write down the Lagrange form of interpolating polynomial.

(b) Write down the Newton form of interpolating polynomial.

A set of data points given in the table below is to be interpolated by a polynomial:

x_j	1	3	4	5
y_j	8	36	32	0

(a) Complete the following general-purpose MatLab program that evaluates the Lagrange polynomial ℓ_j at one or more points.

```
function y = mylagrange(xdp, j, x)
% input:
% xdp   abscissas of data points
% j     evaluate j-th lagrange polynomial
% x     points where polynomial or derivative is evaluated
% (scalar, vector, matrix)
nr_dp = length(xdp);
y = 1;
```

(b) Using the function `mylangrange`, write a script that plots the interpolating polynomial which passes through the given data points on the interval $[0, 6]$. Draw red circles around the data points.

Let $\rho_{n-1}(x)$ be defined by

$$\rho_{n-1}(x) = \prod_{j=1}^{n}(x - x_j) = (x - x_1)(x - x_2)\cdots(x - x_n)\,,$$

for a given set of data $\{x_j \mid j \in \mathbb{N}[1, n]\}$. Write a MatLab program which plots $\rho_{n-1}(x)$ on $[-1, 1]$ for

- uniform nodes $x_j = -1 + (j-1)\Delta x$ with $\Delta x = 2/(n-1)$;
- Chebyshev nodes $x_j = -\cos((j - 1/2)\Delta\theta)$ with $\Delta\theta = \pi/n$.

Problem 11. (Error Analysis)

Consider interpolating $f(x) = \sin(\pi x)$ using a polynomial $p_{n-1}(x)$ on the interval $[-1, 1]$ with $n = 5$ uniform nodes, that is, $\{-1, -1/2, 0, 1/2, 1\}$. Using the error theorem for polynomial interpolation, find an upper bound for the error $f(x) - p_{n-1}(x)$.

Problem 12. (Cubic Spline with *Not-A-Knot* Boundary Conditions)

A set of data points $\{(x_i, y_i) \,|\, i = 1, 2, \ldots, n\}$ is interpolated by a cubic spline $p\!\!\!p(x) = p_i(x)$ on $[x_i, x_{i+1}]$ with

$$p_i(x) = c_{i,1} + c_{i,2}(x - x_i)c_{i,3}(x - x_i)^2 + c_{i,4}(x - x_i)^3.$$

Derive the two equations on p. 22 of Lecture 13 slides implementing the *not-a-knot* boundary conditions.

The following set of data points are to be fitted to a straight line $p(x) = c_1 + c_2 x$ via Linear Least Square approximation:

x_j	0	2	4
y_j	1	3	2

(a) Write out the conditions y_j"="$p(x_j)$, for $1 \leqslant j \leqslant 3$, and turn them into a matrix *equation* of the form $\mathbf{y}$"="$X\mathbf{c}$.

(b) Write out the squared 2-norm of the residual $\|\mathbf{r}\|_2^2$ where $\mathbf{r} = X\mathbf{c} - \mathbf{y}$; call it $g(c_1, c_2)$. **DO NOT** simplify your answer.

(c) The function g is minimized at $\mathbf{c}$ where $\nabla g = \mathbf{0}$. Turn this condition into a single matrix equation for $\mathbf{c}$.

(d) Verify that the result of the previous part agrees with the *normal equation* $X^T X\mathbf{c} = X^T\mathbf{y}$.

Problem 14. (Properties of Orthogonal Matrices)

Let $Q \in \mathbb{R}^{n \times n}$ be orthogonal. Show that

(a) $\|Q\mathbf{x}\|_2 = \|\mathbf{x}\|_2$, for all $\mathbf{x} \in \mathbb{R}^n$.

(b) $\|Q\|_2 = 1$.

(c) $\kappa_2(Q) = 1$. (*Hint.* You are allowed to use, without proof, the facts that $Q^{-1} = Q^{\mathrm{T}}$ and Q^{T} is orthogonal.)

Problem 15. (Householder Transformation)

Let $\mathbf{z} \in \mathbb{R}^n$ be given.

(a) Write down the definition of the Householder matrix H associated with $\mathbf{z}$.

(b) Show that H is symmetric and orthogonal.

(c) Show that $\|H\mathbf{z}\|_2 = \|\mathbf{z}\|_2$.

(d) Suppose that $\mathbf{z}$ is stored in MatLab as a column vector, but you do not know its size. Write a script that creates the associated H. Make sure that H is computed stably, avoiding any potential catastrophic cancellation. Since it is a script, no local function is to be defined.

Problem 16. (Pseudoinverse)

Let $A \in \mathbb{R}^{m\times n}$ with $m \geqslant n$. Using the QR factorization $A = QR$, write down its pseudoinverse $A^\dagger$.

Problem 17. (Polynomial Evaluation for Matrices)

Let $p(z) = c_1 + c_2 z + \cdots + c_{n+1} z^n$. The value of p for a matrix argument is defined as

$$p(A) = c_1 I + c_2 A + \cdots + c_{n+1} A^n.$$

Show that if A is a square matrix and has an EVD, then $p(A)$ can be found using only evaluations of p at the eigenvalues and two matrix multiplications.

Calculate the singular values of

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ -1 & -1 \end{bmatrix}$$

by solving a 2×2 eigenvalue problem.

Problem 19. (SVD and the 2-Norm)

Let $A \in \mathbb{R}^{n \times n}$. Show that

(a) A and A^{T} have the same singular values.

(b) $\|A\|_2 = \left\|A^{\mathrm{T}}\right\|_2$.

Problem 20. (Rayleigh Quotient)

Let

$$A = \begin{bmatrix} 3 & -1 \\ -2 & 0 \end{bmatrix}.$$

(a) Write out $R_A(\mathbf{x})$ explicitly as a function of x_1 and x_2.

(b) Find $R_A(\mathbf{x})$ for $x_1 = 1$, $x_2 = 2$.

(c) Find the gradient vector $\nabla R_A(\mathbf{x})$.

(d) Show that the gradient vector is zero when $x_1 = 1$, $x_2 = 2$.