Updated on July 30, 2020

Review Problems for Final Exam (Solutions) — 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:

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

Solution:

Step 1: Partial pivoting and row operations
[**P**] The pivot element is found on the third row. So swap $\mathcal{R}_1 \leftrightarrow \mathcal{R}_3$:

$$\xrightarrow{\text{row swap}} \begin{bmatrix} -2 & 4 & 2 \\ 0 & 2 & 2 \\ 1 & -2 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 1 \end{bmatrix}$$

[**Z**] Introduce zeros below the $(1,1)$-position by $\mathcal{R}_3 \to \mathcal{R}_3 + (1/2)\mathcal{R}_1$:

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

Fortunately, we have already obtained a desired upper triangular system.

Step 2: Backward substitution
We find that $x_3 = 1$ from the third equation, then $x_2 = 1$ from the second, and finally $x_1 = 2$. Thus

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix}.$$

Problem 2. (Permutation Matrices)

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.

Solution: Let

$$P = P(1,4) = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Since left-multiplication and right-multiplication (by a permutation matrix) acts on rows and columns, respectively,

(a) PA

(b) AP

(c) PAP

(d) For each previous part,

```
>> A([1 4], :) = A([4 1], :);                                   % (a)
>> A(:, [1 4]) = A(:, [4 1]);                                   % (b)
>> A(:, [1 4]) = A(:, [4 1]); A([1 4], :) = A([4 1], :);        % (c)
```

(e) Let

$$P = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Note that we obtain this by permuting rows of the identity matrix I as described. Then left-multiplication by P results in the desired effect. Observe that this action shifts the first four rows cyclically, leaving the fifth unchanged. Thus, after four multiplication, all rows are back to their original positions. Thus $P^k = I$ for any $k = 4n$ with $n \in \mathbb{N}$ and so $k = 4$ is the smallest such positive integer. It is clear that this cyclic permutation can be broken into a series of simple swaps:

$$(\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3, \mathcal{R}_4) \xrightarrow{P(1,2)} (\mathcal{R}_2, \mathcal{R}_1, \mathcal{R}_3, \mathcal{R}_4) \xrightarrow{P(2,3)} (\mathcal{R}_2, \mathcal{R}_3, \mathcal{R}_1, \mathcal{R}_4) \xrightarrow{P(3,4)} (\mathcal{R}_2, \mathcal{R}_3, \mathcal{R}_4, \mathcal{R}_1).$$

Hence,

$$PA = P(3,4)P(2,3)P(1,2)A.$$

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.

Solution:

(a) Backward substitution for the upper triangular system $U\mathbf{x} = \mathbf{y}$ yields

$$\begin{aligned} x_n &= \frac{y_n}{U_{n,n}}\,, \\ x_i &= \frac{y_i - \sum_{j=i+1}^{n} U_{i,j}x_j}{U_{i,i}} \quad \text{for } i = n-1, n-2, \ldots, 2, 1\,, \end{aligned}$$

which can be coded as

```
%% function m-file: back_subs
function x = back_subs(U, y)
  n = length(y);
  x = zeros(size(y));
  x(n) = y(n)/U(n,n);
  for i = n-1:-1:1
      x(i) = ( y(i) - U(i, i+1:end))*x(i+1:end) )/U(i,i);
  end
end
```

(b) It takes only 1 division to calculate x_n. For each $i \in \mathbb{N}[1, n-1]$, the operations involved in calculating x_i are

- $(n-i)$ multiplications to form $U_{i,j}x_j$ for $j = i+1, \ldots, n$;
- $(n-i-1)$ additions to carry out $\sum_{j=i+1}^{n}$;
- 1 subtraction between y_i and previously calculated sum;
- 1 division by $U_{i,i}$.

Thus, the total number of floating-point operations is

$$1 + \sum_{i=1}^{n-1}(2n-2i+1) = 1 + 2\sum_{i=1}^{n-1} n - 2\sum_{i=1}^{n-1} i + \sum_{i=1}^{n-1} 1 \sim 1 + 2n^2 - 2\frac{n^2}{2} + n \sim n^2,$$

for large n.

Note: For large n, $\displaystyle\sum_{i=1}^{n} i^p \sim \frac{n^{p+1}}{p+1}$, which resembles the power rule of integrals $\displaystyle\int x^p \, dx = \frac{x^{p+1}}{p+1} + C$.

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.)

Solution: It is sufficient to show that $G_jG_j^{-1} = G_j^{-1}G_j = I$. Using the given alternate expressions, we write $G_jG_j^{-1}$ as

$$(I + a_{i,j}\mathbf{e}_i\mathbf{e}_j^{\mathrm{T}})(I - a_{i,j}\mathbf{e}_i\mathbf{e}_j^{\mathrm{T}}) = I + \cancel{a_{i,j}\mathbf{e}_i\mathbf{e}_j^{\mathrm{T}}} - \cancel{a_{i,j}\mathbf{e}_i\mathbf{e}_j^{\mathrm{T}}} - a_{i,j}^2\mathbf{e}_i\mathbf{e}_j^{\mathrm{T}}\mathbf{e}_i\mathbf{e}_j^{\mathrm{T}} = I - a_{i,j}^2\mathbf{e}_i\mathbf{e}_j^{\mathrm{T}}\mathbf{e}_i\mathbf{e}_j^{\mathrm{T}}.$$

Note that the last term vanishes:

$$\mathbf{e}_i\left(\mathbf{e}_j^{\mathrm{T}}\mathbf{e}_i\right)\mathbf{e}_j^{\mathrm{T}} = \mathbf{e}_i\delta_{j,i}\mathbf{e}_j^{\mathrm{T}} = 0$$

since $j \neq i$. So $G_jG_j^{-1} = I$. We can show similarly that $G_j^{-1}G_j = I$, which completes the proof.

Note: The inverse of a general Gaussian transformation matrix of the form

$$G_j = I + \sum_{i=j+1}^{n} a_{i,j}\mathbf{e}_i\mathbf{e}_j^{\mathrm{T}} = \begin{bmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & a_{j+1,j} & \ddots & & \\ & & \vdots & & \ddots & \\ & & a_{n,j} & & & 1 \end{bmatrix},$$

is obtained simply by flipping signs of the off-diagonal terms, *i.e.*,

$$G_j = I - \sum_{i=j+1}^{n} a_{i,j}\mathbf{e}_i\mathbf{e}_j^{\mathrm{T}} = \begin{bmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & -a_{j+1,j} & \ddots & & \\ & & \vdots & & \ddots & \\ & & -a_{n,j} & & & 1 \end{bmatrix}.$$

Would you be able to show this?

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.)

Solution:

1. **Pivot:** $\mathcal{R}_1 \leftrightarrow \mathcal{R}_2$
$$\underbrace{\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{P(1,2)} \begin{bmatrix} 2 & -1 & 3 \\ 10 & -7 & 10 \\ -6 & 4 & -5 \end{bmatrix} = \begin{bmatrix} 10 & -7 & 10 \\ 2 & -1 & 3 \\ -6 & 4 & -5 \end{bmatrix} = P(1,2)A$$
2. **Zero:** $\mathcal{R}_2 \to \mathcal{R}_2 - {}^1\!/_5\mathcal{R}_1$ and $\mathcal{R}_3 \to \mathcal{R}_3 + {}^3\!/_5\mathcal{R}_1$
$$\underbrace{\begin{bmatrix} 1 & 0 & 0 \\ -\frac{1}{5} & 1 & 0 \\ \frac{3}{5} & 0 & 1 \end{bmatrix}}_{G_1} \begin{bmatrix} 10 & -7 & 10 \\ 2 & -1 & 3 \\ -6 & 4 & -5 \end{bmatrix} = \begin{bmatrix} 10 & -7 & 10 \\ 0 & \frac{2}{5} & 1 \\ 0 & -\frac{1}{5} & 1 \end{bmatrix} = G_1P(1,2)A$$
3. **Pivot:** No pivoting is necessary on the second column.
4. **Zero:** $\mathcal{R}_3 \to \mathcal{R}_3 + {}^1\!/_2\mathcal{R}_2$
$$\underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \frac{1}{2} & 1 \end{bmatrix}}_{G_2} \begin{bmatrix} 10 & -7 & 10 \\ 0 & \frac{2}{5} & 1 \\ 0 & -\frac{1}{5} & 1 \end{bmatrix} = \underbrace{\begin{bmatrix} 10 & -7 & 10 \\ 0 & \frac{2}{5} & 1 \\ 0 & 0 & \frac{3}{2} \end{bmatrix}}_{U} = G_2G_1P(1,2)A$$

Letting $P = P(1,2)$, the above process is summarized as

$$G_2G_1P(1,2)A = U \qquad \longrightarrow \qquad PA = \underbrace{G_1^{-1}G_2^{-1}}_{L}U\,.$$

Using the result of the previous problem, we see that

$$G_1^{-1}G_2^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{5} & 1 & 0 \\ -\frac{3}{5} & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{1}{2} & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{5} & 1 & 0 \\ -\frac{3}{5} & -\frac{1}{2} & 1 \end{bmatrix} = L\,.$$

Therefore,

$$P = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad L = \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{5} & 1 & 0 \\ -\frac{3}{5} & -\frac{1}{2} & 1 \end{bmatrix}, \quad U = \begin{bmatrix} 10 & -7 & 10 \\ 0 & \frac{2}{5} & 1 \\ 0 & 0 & \frac{3}{2} \end{bmatrix}.$$

Problem 6. (Step-By-Step LDLT Factorization)

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.

Solution: See `lec10_special_matrices_outline.mlx`.

Let

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

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

Solution: We use the following convenient formulas for matrix p-norms for $p = 1, \infty$:

$$\|A\|_1 = \max_{1 \leqslant j \leqslant 2} \left\{ \sum_{i=1}^{2} |A_{i,j}| \right\} = \max\{1, 5\} = 5,$$

$$\|A\|_\infty = \max_{1 \leqslant i \leqslant 2} \left\{ \sum_{j=1}^{2} |A_{i,j}| \right\} = \max\{3, 3\} = 3.$$

The calculation of Frobenius norm is very straightfoward:

$$\|A\|_F = \sqrt{\sum_{i=1}^{2} \sum_{j=1}^{2} |A_{i,j}|} = \sqrt{1^2 + 2^2 + 3^3} = \sqrt{14}.$$

To compute the 2-norm of A. We first find that

$$A^{\mathrm{T}} A = \begin{bmatrix} 1 & 0 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 0 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 2 & 13 \end{bmatrix}$$

has two real positive eigenvalues:

$$\left\{ \begin{aligned} \det\left(\lambda I - A^{\mathrm{T}} A\right) &= \det \begin{bmatrix} \lambda - 1 & -2 \\ -2 & \lambda - 13 \end{bmatrix} \\ &= (\lambda - 1)(\lambda - 13) - 4 \\ &= \lambda^2 - 14\lambda + 9. \end{aligned} \right\} \implies \lambda = 7 \pm 2\sqrt{10}.$$

Therefore,

$$\|A\|_2 = \sqrt{\lambda_{\max}(A^{\mathrm{T}} A)} = \sqrt{7 + 2\sqrt{10}}.$$

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

Solution: Note that $\|A\|_1$ is equal to the 1-norm of the second column vector of A. Since $A\mathbf{e}_2$ is the second column of A and since $\mathbf{e}_2$ is a unit vector, we conclude that $\mathbf{x} = \pm \mathbf{e}_2$ satisfies the required conditions.

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

```
function x = 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`.)

Solution:

```
function x = 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
    if j == 1
        x = max(sum(abs(A), 1));
    elseif j == 2
        x = sqrt(max(eig(A'*A)));
    elseif strcmp(j, 'inf')
        x = max(sum(abs(A), 2));
    elseif strcmp(j, 'fro')
        x = sqrt(sum(sum(abs(A).^2)));
    else
        error('Invalid second input.');
    end
end
```

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.

Solution:

(a) First, we find the Lagrange polynomials associated with the given data set:

$$\begin{aligned}
\ell_1(x) &= \frac{(x-3)(x-5)(x-8)}{(2-3)(2-5)(2-8)} = -\frac{1}{18}(x-3)(x-5)(x-8)\,,\\
\ell_2(x) &= \frac{(x-2)(x-5)(x-8)}{(3-2)(3-5)(3-8)} = \frac{1}{10}(x-2)(x-5)(x-8)\,,\\
\ell_3(x) &= \frac{(x-2)(x-3)(x-8)}{(5-2)(5-3)(5-8)} = -\frac{1}{18}(x-2)(x-3)(x-8)\,,\\
\ell_4(x) &= \frac{(x-2)(x-3)(x-5)}{(8-2)(8-3)(8-5)} = \frac{1}{90}(x-2)(x-3)(x-5)\,.
\end{aligned}$$

Then the interpolating polynomial is

$$\begin{aligned}
p_3(x) &= 3\ell_1(x) - 2\ell_2(x) + 12\ell_3(x) + 3\ell_4(x)\\
&= -\frac{1}{6}(x-3)(x-5)(x-8) - \frac{1}{5}(x-2)(x-5)(x-8)\\
&\qquad - \frac{2}{3}(x-2)(x-3)(x-8) + \frac{1}{30}(x-2)(x-3)(x-5)\,.
\end{aligned}$$

(b) We calculate Newton's divided differences using the schematic introduced in class:

$$\begin{array}{cccc}
2 : 3 = 3 & & & \\
 & \dfrac{-2-3}{3-2} = -5 & & \\
3 : -2 = -2 & & \dfrac{7+5}{5-2} = 4 & \\
 & \dfrac{12+2}{5-3} = 7 & & \dfrac{-2-4}{8-2} = -1 \\
5 : 12 = 12 & & \dfrac{-3-7}{8-3} = -2 & \\
 & \dfrac{3-12}{8-5} = -3 & & \\
8 : 3 = 3 & & &
\end{array}$$

With respect to the Newton's basis $\{1, x-2, (x-2)(x-3), (x-2)(x-3)(x-5)\}$, the interpolating polynomial is written as

$$p_3(x) = 3 - 5(x-2) + 4(x-2)(x-3) - (x-2)(x-3)(x-5)\,.$$

Problem 9. (Lagrange and Newton Polynomial Interpolation (2))

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;
```

Solution:

```
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;                          % or y = ones(size(x))
  for k = 1:nr_dp
  if k ~= j
    y = y .* (x - xdp(k))/(xdp(j) - xdp(k));
  end
end
```

(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.

Solution:

```
xdp = [1 3 4 5];
ydp = [8 36 32 0];
x = linspace(0, 6, 100);
y = zeros(size(x));
for j = 1:length(xdp)
y = y + ydp(j)*mylangrange(xdp, j, x);
end
plot(x, y, xdp, ydp, 'ro')
```

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 $\Delta\theta = \pi/n$.

Solution:

```
%% script m-file: node_comparison
clear, close all
n = 11;

%% generate nodes:
xdp_unif = -1 + 2*(0:n-1)/(n-1);  % or simply use: linspace(-1, 1, n);
xdp_cheb = -cos(((1:n) - .5)*pi/n);

%% polynomial evaluation:
xs = linspace(-1, 1, 1001)';
ys_unif = prod(xs - xdp_unif, 2);
ys_cheb = prod(xs - xdp_cheb, 2);

%% plot
subplot(2,1,1)
plot(xs, ys_unif);
hold on;
plot([-1 1], [0 0], 'k',...       % draw x-axis
     xdp_unif, zeros(1,n), 'ro'); % circle zeros
y_endpoints = ylim();             % grab vert. limits
grid on, title('Equispaced nodes')

subplot(2,1,2)
plot(xs, ys_cheb);
hold on;
plot([-1 1], [0 0], 'k',...       % draw x-axis
     xdp_cheb, zeros(1,n), 'ro'); % circle zeros
ylim(y_endpoints)                 % vertical scaling
grid on, title('Chebyshev nodes')
```

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 $f(x) - p_{n-1}(x)$.

Solution: For this problem, the following result specialized for the case of uniform grids is useful.

Polynomial Interpolation Error Bound – Uniform Grid

For n evenly-distributed data points $a = x_1 < x_2 < \cdots < x_n = b$, *i.e.*, $x_j = a + (j-1)\Delta x$ with $\Delta x = (b-a)/(n-1)$,

$$\|f - p_{n-1}\|_\infty \leqslant \frac{\|f^{(n)}\|_\infty}{4n} (\Delta x)^n.$$

In this problem, $n = 5$ and $\Delta x = 1/2$. Furthermore, since $f(x) = \sin(\pi x)$, $f'(x) = \pi \cos(\pi x)$, $\ldots$, $f^{(5)}(x) = \pi^5 \cos(\pi x)$,

$$\left\|f^{(5)}\right\|_\infty = \max_{[-1,1]} \left|f^{(5)}(x)\right| = \pi^5.$$

Therefore, using the estimate above, we obtain

$$\|f - p_{n-1}\|_\infty \leqslant \frac{\pi^5}{4 \cdot 5} \left(\frac{1}{2}\right)^5 = \frac{\pi^5}{640}.$$

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

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

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

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

Solution: Recall that *not-a-knot* boundary conditions are used when nothing is known about the endpoints. They require that $p_1 \equiv p_2$ and $p_{n-2} \equiv p_{n-1}$; in words, the first and the last interior breakpoints, x_2 and x_{n-1}, are not actually breakpoints. Since these pairs are already required to have matching derivatives up to the second order at x_2 and x_{n-1}, it is only necessary to impose equality of the third derivatives at these breakpoints:

$$p_1'''(x_2) = p_2'''(x_2) \quad \text{and} \quad p_{n-2}'''(x_{n-1}) = p_{n-1}'''(x_{n-1})$$

From (☺), note that $p_i'''(x) = 6c_{i,4}$. So the constraints are written out as

$$c_{1,4} = c_{2,4} \quad \text{and} \quad c_{n-2,4} = c_{n-1,4}.$$

Using the following representation for coefficient $c_{i,4}$ given on p. 16 of Lecture 13 slides

$$c_{i,4} = \frac{\sigma_i + \sigma_{i+1} - 2y[x_i, x_{i+1}]}{(\Delta x_i)^2},$$

the first constraint $c_{1,4} = c_{2,4}$ can be rewritten as

$$\frac{\sigma_1 + \sigma_2 - 2y[x_1, x_2]}{(\Delta x_1)^2} = \frac{\sigma_2 + \sigma_3 - 2y[x_2, x_3]}{(\Delta x_2)^2}.$$

Multiplying both sides by $(\Delta x_1)^2 (\Delta x_2)^2$ and collecting terms involving σ's on the left-hand side, we have

$$(\Delta x_2)^2 \sigma_1 + \left((\Delta x_2)^2 - (\Delta x_1)^2\right) \sigma_2 - (\Delta x_1)^2 \sigma_3 = 2\left(y[x_1, x_2]\,(\Delta x_2)^2 - y[x_2, x_3]\,(\Delta x_1)^2\right).$$

This is precisely the first equation appearing on p. 22 of Lecture 13.

Similarly, the second constraint $c_{n-2,4} = c_{n-1,4}$ can be written as

$$\frac{\sigma_{n-2} + \sigma_{n-1} - 2y[x_{n-2}, x_{n-1}]}{(\Delta x_{n-2})^2} = \frac{\sigma_{n-1} + \sigma_n - 2y[x_{n-1}, x_n]}{(\Delta x_{n-1})^2},$$

which then can be rearranged into

$$\begin{aligned}(\Delta x_{n-1})^2 \sigma_{n-2} + \left((\Delta x_{n-1})^2 - (\Delta x_{n-2})^2\right) \sigma_{n-1} - (\Delta x_{n-2})^2 \sigma_n \\ = 2\left(y[x_{n-2}, x_{n-1}]\,(\Delta x_{n-1})^2 - y[x_{n-1}, x_n]\,(\Delta x_{n-2})^2\right).\end{aligned}$$

If both sides are negated, it becomes the second equation to be shown.

Problem 13. (Linear Least Squares Approximation and Normal Equation)

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 $\mathbf{y}$"="$X\mathbf{c}$.

> **Solution:** Writing down the interpolating conditions, we have
>
> $$\left\{\begin{array}{c} y_1 = p(x_1) \\ y_2 = p(x_2) \\ y_3 = p(x_3) \end{array}\right\} \implies \left\{\begin{array}{c} 1 = c_1 + c_2 \cdot 0 \\ 3 = c_1 + c_2 \cdot 2 \\ 2 = c_1 + c_2 \cdot 4 \end{array}\right\} \implies \underbrace{\begin{bmatrix} 1 \\ 3 \\ 2 \end{bmatrix}}_{=:\mathbf{y}} = \underbrace{\begin{bmatrix} 1 & 0 \\ 1 & 2 \\ 1 & 4 \end{bmatrix}}_{=:X} \underbrace{\begin{bmatrix} c_1 \\ c_2 \end{bmatrix}}_{=:\mathbf{c}}$$

(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.

> **Solution:** From the previous part,
>
> $$\mathbf{r} = X\mathbf{c} - \mathbf{y} = \begin{bmatrix} c_1 - 1 \\ c_1 + 2c_2 - 3 \\ c_1 + 4c_2 - 2 \end{bmatrix}.$$
>
> Thus the squared 2-norm of the residual is given by
>
> $$\|\mathbf{r}\|_2^2 = (c_1 - 1)^2 + (c_1 + 2c_2 - 3)^2 + (c_1 + 4c_2 - 2)^2 =: g(c_1, c_2).$$

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

> **Solution:** From the previous part, $g(c_1, c_2) = (c_1 - 1)^2 + (c_1 + 2c_2 - 3)^2 + (c_1 + 4c_2 - 2)^2$. Setting $\nabla g = \mathbf{0}$, we obtain the following two equations:
>
> $$\frac{\partial g}{\partial c_1} = 2(c_1 - 1) + 2(c_1 + 2c_2 - 3) + 2(c_1 + 4c_2 - 2) = 0$$
> $$\frac{\partial g}{\partial c_2} = 4(c_1 + 2c_2 - 3) + 8(c_1 + 4c_2 - 2) = 0$$
>
> Dividing by 2 and collecting like terms, they simplify to
>
> $$\left\{\begin{array}{r} 3c_1 + 6c_2 = 6 \\ 6c_1 + 20c_2 = 14 \end{array}\right\} \implies \begin{bmatrix} 3 & 6 \\ 6 & 20 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} 6 \\ 14 \end{bmatrix}.$$

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

Solution: By simple matrix calculation,

$$X^{\mathrm{T}}X = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 2 & 4 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 2 \\ 1 & 4 \end{bmatrix} = \begin{bmatrix} 3 & 6 \\ 6 & 20 \end{bmatrix} \quad \text{and} \quad X^{\mathrm{T}}\mathbf{y} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 2 & 4 \end{bmatrix} \begin{bmatrix} 1 \\ 3 \\ 2 \end{bmatrix} = \begin{bmatrix} 6 \\ 14 \end{bmatrix}.$$

Thus, we verify that the normal equation $X^{\mathrm{T}}X\mathbf{c} = X^{\mathrm{T}}\mathbf{y}$ is indeed the same as the one obtained in the previous part.

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.)

Solution:

(a) Observe that

$$\|Q\mathbf{x}\|_2^2 = (Q\mathbf{x})^{\mathrm{T}}(Q\mathbf{x}) = \mathbf{x}^{\mathrm{T}} \underbrace{Q^{\mathrm{T}}Q}_{=I} \mathbf{x} = \mathbf{x}^{\mathrm{T}}\mathbf{x} = \|\mathbf{x}\|_2^2.$$

Note that $Q^{\mathrm{T}}Q = I$ due to the assumed orthogonality of Q. Taking square root, we confirm that multiplication by Q preserves the 2-norm.

(b) By definition and by the previous result, we note that

$$\|Q\|_2 = \max_{\|\mathbf{x}\|_2=1} \|Q\mathbf{x}\|_2 = \max_{\|\mathbf{x}\|_2=1} \|\mathbf{x}\|_2 = 1.$$

(c) Since $Q^{-1} = Q^{\mathrm{T}}$ is also orthogonal, $\|Q^{-1}\|_2 = 1$ by part (b). It follows that

$$\kappa_2(Q) = \|Q\|_2 \|Q^{-1}\|_2 = 1.$$

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 $\mathbf{z}$.

> **Solution:** The Householder matrix H associated with $\mathbf{z}$ is defined by
>
> $$H = I - \frac{2}{\mathbf{v}^{\mathrm{T}}\mathbf{v}}\mathbf{v}\mathbf{v}^{\mathrm{T}},$$
>
> where $\mathbf{v} = \pm \|\mathbf{z}\|_2 \, \mathbf{e}_1 - \mathbf{z}$. Note that H is an $(n \times n)$ square matrix.

(b) Show that H is symmetric and orthogonal.

> **Solution:** Since
>
> $$\begin{aligned} H^{\mathrm{T}} &= \left(I - 2\frac{\mathbf{v}\mathbf{v}^{\mathrm{T}}}{\mathbf{v}^{\mathrm{T}}\mathbf{v}} \right)^{\mathrm{T}} = I^{\mathrm{T}} - \underbrace{\frac{2}{\mathbf{v}^{\mathrm{T}}\mathbf{v}}}_{\text{scalar}} \left(\mathbf{v}\mathbf{v}^{\mathrm{T}} \right)^{\mathrm{T}} \\ &= I - \frac{2}{\mathbf{v}^{\mathrm{T}}\mathbf{v}} \left(\mathbf{v}^{\mathrm{T}} \right)^{\mathrm{T}} \mathbf{v}^{\mathrm{T}} = I - \frac{2}{\mathbf{v}^{\mathrm{T}}\mathbf{v}}\mathbf{v}\mathbf{v}^{\mathrm{T}} = H, \end{aligned}$$
>
> H is symmetric. Consequently,
>
> $$\begin{aligned} H^{\mathrm{T}}H = H^2 &= \left(I - \frac{2}{\mathbf{v}^{\mathrm{T}}\mathbf{v}}\mathbf{v}\mathbf{v}^{\mathrm{T}} \right)^2 \\ &= I - \frac{4}{\mathbf{v}^{\mathrm{T}}\mathbf{v}}\mathbf{v}\mathbf{v}^{\mathrm{T}} + \frac{4}{(\mathbf{v}^{\mathrm{T}}\mathbf{v})^2} \left(\mathbf{v}\mathbf{v}^{\mathrm{T}} \right) \left(\mathbf{v}\mathbf{v}^{\mathrm{T}} \right) \\ &= I - \frac{4}{\mathbf{v}^{\mathrm{T}}\mathbf{v}}\mathbf{v}\mathbf{v}^{\mathrm{T}} + \frac{4}{(\mathbf{v}^{\mathrm{T}}\mathbf{v})^2}\mathbf{v} \left(\mathbf{v}^{\mathrm{T}}\mathbf{v} \right) \mathbf{v}^{\mathrm{T}} \\ &= I - \frac{4}{\mathbf{v}^{\mathrm{T}}\mathbf{v}}\mathbf{v}\mathbf{v}^{\mathrm{T}} + \frac{4}{\mathbf{v}^{\mathrm{T}}\mathbf{v}}\mathbf{v}\mathbf{v}^{\mathrm{T}} = I, \end{aligned}$$
>
> which shows that H is orthogonal.

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

> **Solution:** Observe that
>
> $$\|H\mathbf{z}\|_2^2 = (H\mathbf{z})^{\mathrm{T}} (H\mathbf{z}) = \mathbf{z}^{\mathrm{T}} \underbrace{H^{\mathrm{T}}H}_{=I} \mathbf{z} = \mathbf{z}^{\mathrm{T}}\mathbf{z} = \|\mathbf{z}\|_2^2 \, .$$
>
> (Note the use of orthogonality of H.) Taking square roots, we obtain the requested equality.

(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.

Solution: To compute H stably, we need to choose a proper sign in forming $\mathbf{v}$ [a]:

$$\mathbf{v} = \begin{cases} -\left\|\mathbf{z}\right\|_2 \mathbf{e}_1 - \mathbf{z}, & \text{if } z_1 \geqslant 0; \\ \quad \left\|\mathbf{z}\right\|_2 \mathbf{e}_1 - \mathbf{z}, & \text{if } z_1 < 0. \end{cases}$$

This way, we can avoid a potential catastrophic cancellation when $|z_1| \approx \left\|\mathbf{z}\right\|_2$.

```
n = length(z);
I = eye(n);
normz = norm(z);
if z(1) >= 0
  v = -normz*I(:,1) - z;
else
  v =  normz*I(:,1) - z;
end
H = I - 2/(v' * v) * (v * v');
```

[a] Compare this against p. 13 of Lecture 16 slides. Both are correct. Can you reason why?

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 $A^\dagger$.

Solution: Recall that $A^\dagger = \left(A^{\mathrm{T}}A\right)^{-1}A^{\mathrm{T}}$. So

$$\begin{aligned} A^\dagger &= \Big(R^{\mathrm{T}}\underbrace{Q^{\mathrm{T}}Q}_{=I}R\Big)^{-1}R^{\mathrm{T}}Q^{\mathrm{T}} \\ &= \left(R^{\mathrm{T}}R\right)^{-1}R^{\mathrm{T}}Q^{\mathrm{T}}. \end{aligned} \tag{$\star$}$$

Since $R \in \mathbb{R}^{m\times n}$, one must not hastily jump to writing $\left(R^{\mathrm{T}}R\right)^{-1} = R^{-1}R^{-\mathrm{T}}$. Since R is upper triangular, $R^{\mathrm{T}}R$ can be equivalently written as

$$R^{\mathrm{T}}R = \left[\begin{array}{c|c} \widehat{R}^{\mathrm{T}} & \mathbf{0}^{\mathrm{T}} \end{array}\right]\left[\begin{array}{c} \widehat{R} \\ \hline \mathbf{0} \end{array}\right] = \widehat{R}^{\mathrm{T}}\widehat{R},$$

where $\widehat{R} \in \mathbb{R}^{n\times n}$ is a square upper triangular matrix and $\mathbf{0} \in \mathbb{R}^{(m-n)\times n}$ is a zero matrix. Likewise, we can rewrite $R^{\mathrm{T}}Q^{\mathrm{T}}$ as

$$R^{\mathrm{T}}Q^{\mathrm{T}} = \left[\begin{array}{c|c} \widehat{R}^{\mathrm{T}} & \mathbf{0}^{\mathrm{T}} \end{array}\right]\left[\begin{array}{c} \widehat{Q}^{\mathrm{T}} \\ \hline \widetilde{Q}^{\mathrm{T}} \end{array}\right] = \widehat{R}^{\mathrm{T}}\widehat{Q}^{\mathrm{T}},$$

where $\widehat{Q} \in \mathbb{R}^{m\times n}$ is the $(m\times n)$ submatrix of Q containing the first n columns and $\widetilde{Q} \in \mathbb{R}^{m\times(m-n)}$ is the rest of Q. With these, we can further simplify ($\star$) into

$$A^\dagger = \left(\widehat{R}^{\mathrm{T}}\widehat{R}\right)^{-1}\widehat{R}^{\mathrm{T}}\widehat{Q}^{\mathrm{T}} = \widehat{R}^{-1}\widehat{Q}^{\mathrm{T}}.$$

Compare this to the formula for $A^\dagger$ derived using the *thin* QR factorization in Lecture 16.

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.

Solution: Let $A \in \mathbb{R}^{k\times k}$ be written as $A = VDV^{-1}$ where

$$V = \left[\begin{array}{c|c|c} \mathbf{v}_1 & \cdots & \mathbf{v}_k \end{array}\right] \in \mathbb{R}^{k\times k}, \qquad \text{(eigenvectors)}$$

and

$$D = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_k \end{bmatrix} \in \mathbb{R}^{k\times k}. \qquad \text{(eigenvalues)}$$

Note that, for any $1 \leqslant j \leqslant n$,

$$A^j = \underbrace{\left(VDV^{-1}\right)\cdots\left(VDV^{-1}\right)}_{j \text{ copies}} = VD^jV^{-1},$$

and so

$$\begin{aligned} p(A) &= c_1 I + c_2 VDV^{-1} + c_3 VD^2V^{-1} + \cdots + c_{n+1} VD^nV^{-1} \\ &= V \underbrace{\left(c_1 I + c_2 D + c_3 D^2 + \cdots + c_{n+1} D^n\right)}_{=p(D)} V^{-1} = Vp(D)V^{-1}. \end{aligned}$$

Observe that $p(D)$ is a diagonal matrix whose (j,j)-entry is given by

$$c_1 + c_2\lambda_j + c_3\lambda_j^2 + \cdots + c_{n+1}\lambda_j^n = p(\lambda_j), \quad 1 \leqslant j \leqslant k.$$

That is,

$$p(D) = \begin{bmatrix} p(\lambda_1) & & \\ & \ddots & \\ & & p(\lambda_k) \end{bmatrix},$$

and so it only requires evaluations of p at the eigenvalues. Once it is constructed, one can form $p(A)$ by multiplying it by V and V^{-1}, 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.

Solution: Recall that the nonzero singular values of A are the square roots of the nonzero eigenvalues of $A^{\mathrm{T}}A$; see Theorem 3 of Lecture 18. First, compute $A^{\mathrm{T}}A$:

$$A^{\mathrm{T}}A = \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ -1 & -1 \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}.$$

Then find its eigenvalues:

$$\left\{ \begin{aligned} \det\left(\lambda I - A^{\mathrm{T}}A\right) &= \begin{vmatrix} \lambda - 2 & -1 \\ -1 & \lambda - 2 \end{vmatrix} \\ &= (\lambda - 2)^2 - 1 \\ &= \lambda^2 - 4\lambda + 3. \end{aligned} \right\} \implies \lambda_1 = 3,\ \lambda_2 = 1.$$

(Note that all eigenvalues are nonnegative; I order them in descending order so that the singular values are ordered properly.)
We conclude that the two singular values of A are

$$\begin{aligned} \sigma_1 &= \sqrt{\lambda_1} = \sqrt{3}, \\ \sigma_2 &= \sqrt{\lambda_2} = 1. \end{aligned}$$

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) $\|A\|_2 = \|A^{\mathrm{T}}\|_2$.

Solution:

(a) Suppose that $A = U\Sigma V^{\mathrm{T}}$ is an SVD of A. Then

$$A^{\mathrm{T}} = \left(U\Sigma V^{\mathrm{T}}\right)^{\mathrm{T}} = V\Sigma^{\mathrm{T}}U^{\mathrm{T}} = V\Sigma U^{\mathrm{T}}.$$

Note that $\Sigma^{\mathrm{T}} = \Sigma$ since it is an $(n \times n)$ diagonal matrix. Since U and V are orthogonal matrices, the last factorization is an SVD of A^{T}. In particular, the singular values of A^{T} are the diagonal entries of Σ which are also the singular values of A.

(b) From the previous part, we know that both matrices share the same set of singular values. Since the 2-norm of a matrix is its largest singular value, it follows that $\|A\|_2 = \left\|A^{\mathrm{T}}\right\|_2$.

Problem 20. (Rayleigh Quotient)

Let

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

(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 $x_1 = 1$, $x_2 = 2$.

Solution: Recall that the Rayleigh quotient R_A associated with $A \in \mathbb{R}^{n\times n}$ is defined by

$$R_A(\mathbf{x}) = \frac{\mathbf{x}^{\mathrm{T}} A\mathbf{x}}{\mathbf{x}^{\mathrm{T}}\mathbf{x}}.$$

(a) Let $\mathbf{x} = (x_1, x_2)^{\mathrm{T}}$. Then

$$\begin{aligned} \mathbf{x}^{\mathrm{T}} A\mathbf{x} &= \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} 3 & -2 \\ -2 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad \text{and} \qquad \mathbf{x}^{\mathrm{T}}\mathbf{x} = x_1^2 + x_2^2. \\ &= 3x_1^2 - x_1x_2, \end{aligned}$$

Thus,

$$R_A(\mathbf{x}) = \frac{3x_1^2 - 4x_1x_2}{x_1^2 + x_2^2}.$$

(b) Let $\mathbf{x} = (1, 2)^{\mathrm{T}}$. Then by the expression found above,

$$R_A(\mathbf{x}) = \frac{3(1)^2 - 4(1)(2)}{1^2 + 2^2} = -1.$$

Note. One can show that $\lambda = -1$ is an eigenvalue of A and $\mathbf{v} = (1, 2)^{\mathrm{T}}$ is an associated eigenvector. This part demonstrates the fact that the Rayleigh quotient R_A maps an eigenvector of A into its associated eigenvalue.

(c) Upon partial differentiation, we find that

$$\begin{aligned} \frac{\partial R_A}{\partial x_1} &= \frac{6x_1 - 4x_2}{x_1^2 + x_2^2} - 2x_1 \frac{3x_1^2 - 4x_1x_2}{\left(x_1^2 + x_2^2\right)^2} \\ &= \frac{(6x_1 - 4x_2)(x_1^2 + x_2^2) - 2x_1(3x_1^2 - 4x_1x_2)}{(x_1^2 + x_2^2)^2} \\ &= \frac{4x_1^2x_2 + 6x_1x_2^2 - 4x_2^3}{(x_1^2 + x_2^2)^2} \\ &= 2x_2 \frac{2x_1^2 + 3x_1x_2 - 2x_2^2}{(x_1^2 + x_2^2)^2} \end{aligned}$$

$$\begin{aligned} \frac{\partial R_A}{\partial x_2} &= \frac{-4x_1}{x_1^2 + x_2^2} - 2x_2 \frac{3x_1^2 - 4x_1x_2}{(x_1^2 + x_2^2)^2} \\ &= \frac{-4x_1(x_1^2 + x_2^2) - 2x_2(3x_1^2 - 4x_1x_2)}{(x_1^2 + x_2^2)^2} \\ &= \frac{-4x_1^3 - 6x_1^2x_2 + 4x_1x_2^2}{(x_1^2 + x_2^2)^2} \\ &= -2x_1 \frac{2x_1^2 + 3x_1x_2 - 2x_2^2}{(x_1^2 + x_2^2)^2} \end{aligned}$$

So

$$\nabla R_A(\mathbf{x}) = \frac{2(2x_1^2 + 3x_1x_2 - 2x_2^2)}{(x_1^2 + x_2^2)^2} \begin{bmatrix} x_2 \\ -x_1 \end{bmatrix}.$$

(d) Note that the common factor $2x_1^2 + 3x_1x_2 - 2x_2^2$ vanishes when $x_1 = 1$ and $x_2 = 2$. It follows that the gradient ∇R_A is zero with those values.