MH5200 Advanced Investigations in Linear Algebra I

Problem Sheet 3– Problems & Solutions

Academic Year 2025/2026, Semester 1

Quantitative Research Society @NTU

September 14, 2025

Overview of This Problem Sheet

Where pedagogically helpful, we:

- expand arguments into explicit step-by-step derivations;
- add proof-structure hints (e.g. "consider the contrapositive", "reduce to eigenbasis", "apply spectral theorem", "use Schur complement");
- highlight advanced techniques frequently used in MH5200 (block matrix methods, invariant subspaces, polynomial functional calculus, spectral decompositions, singular value arguments, etc.).

Structure of the sheet.

- **Problem 1:** Quartic polynomial interpolation as a linear system; rank and under-determined family of solutions.
- **Problem 2:** Trace as Frobenius inner product; symmetry, positivity, and cyclicity of trace.
- **Problem 3:** Magic matrix conditions in matrix form; row/column sums and diagonal sums via eigenvalue and trace constraints.
- **Problem 4:** Symptom incidence matrix; interpretations of products as counts and co-occurrence Gram matrices.
- **Problem 5:** Circular shift permutation matrix; action on vectors and finite-order property $C^5 = I$.
- **Problem 6:** Linear operators on a 3×3 image; permutation and averaging matrices for basic geometric transforms.

Problem 1

Problem

Let $c = (c_1, c_2, c_3, c_4, c_5)^T$ represent the coefficients of the quartic polynomial

$$p(x) = c_1 + c_2 x + c_3 x^2 + c_4 x^3 + c_5 x^4.$$

Express the conditions

$$p(0) = 0, \quad p'(0) = 0, \quad p(1) = 1, \quad p'(1) = 0,$$

as a set of linear equations $Ac = b$. Is the system under-determined, over-determined, or uniquely determined?

Solution

Method 1: Direct Evaluation and Derivatives

We compute

$$p(0) = c_1, \qquad p'(x) = c_2 + 2c_3 x + 3c_4 x^2 + 4c_5 x^3, \qquad p'(0) = c_2.$$

At $x = 1$,

$$p(1) = c_1 + c_2 + c_3 + c_4 + c_5, \qquad p'(1) = c_2 + 2c_3 + 3c_4 + 4c_5.$$

The four conditions become the linear equations

$$\begin{aligned} p(0) = 0 &\iff c_1 = 0, \\ p'(0) = 0 &\iff c_2 = 0, \\ p(1) = 1 &\iff c_1 + c_2 + c_3 + c_4 + c_5 = 1, \\ p'(1) = 0 &\iff c_2 + 2c_3 + 3c_4 + 4c_5 = 0. \end{aligned}$$

In matrix form,

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 4 \end{bmatrix}, \qquad c = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{bmatrix}, \qquad b = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix},$$

so that $Ac = b$ encodes all four conditions.

To determine whether the system is under-/over-/uniquely determined, we examine $\operatorname{rank}(A)$. The first two rows fix c_1 and c_2 and are clearly independent. Restricting to the last three columns (for c_3, c_4, c_5), we have the 2×3 submatrix

$$\begin{bmatrix} 1 & 1 & 1 \\ 2 & 3 & 4 \end{bmatrix},$$

which has rank 2 (its two rows are not multiples of each other). Thus

$$\mathrm{rank}(A) = 2 + 2 = 4.$$

There are 5 unknowns and 4 independent equations, so the solution space has dimension $5 - 4 = 1$: there is a one-parameter family of solutions. The system is therefore *under-determined*.

Method 2: Factorisation as an Interpolation Problem

The conditions at $x = 0$,

$$p(0) = 0, \quad p'(0) = 0,$$

imply that $x = 0$ is a root of multiplicity at least 2. Hence we can factor

$$p(x) = x^2 q(x),$$

where q is a polynomial of degree at most 2:

$$q(x) = a + bx + cx^2.$$

The remaining conditions at $x = 1$ become

$$p(1) = q(1) = 1, \qquad p'(1) = 0.$$

Now

$$q(1) = a + b + c = 1,$$

and

$$p'(x) = 2xq(x) + x^2 q'(x) \quad \Rightarrow \quad p'(1) = 2q(1) + q'(1) = 0.$$

Since $q(1) = 1$, we get

$$2 \cdot 1 + q'(1) = 0 \quad \Rightarrow \quad q'(1) = -2.$$

But $q'(x) = b + 2cx$, so

$$q'(1) = b + 2c = -2.$$

We have two linear equations for three unknowns a, b, c:

$$\begin{cases} a + b + c = 1, \\ b + 2c = -2. \end{cases}$$

Solving in terms of the free parameter c:

$$b = -2 - 2c, \qquad a = 1 - b - c = 1 - (-2 - 2c) - c = 3 + c.$$

Thus

$$q(x) = (3 + c) + (-2 - 2c)x + cx^2,$$

and

$$p(x) = x^2 q(x) = x^2 \left[(3 + c) + (-2 - 2c)x + cx^2\right],$$

which is a one-parameter family of quartics satisfying the four conditions. This again shows that the system is under-determined with one degree of freedom.

Problem 2

Problem

Let $\operatorname{tr}(A)$ denote the trace of a square matrix A. For real $m \times n$ matrices $A = (a_{ij})$ and $B = (b_{ij})$, prove:

(a) $\operatorname{tr}(A^T B) = \sum_{i=1}^m \sum_{j=1}^n a_{ij} b_{ij}$.

(b) $\operatorname{tr}(A^T B) = \operatorname{tr}(B^T A)$.

(c) $\operatorname{tr}(A^T A) \geq 0$, with equality iff $A = 0$.

(d) $\operatorname{tr}(A^T B) = \operatorname{tr}(BA^T)$, even if dimensions differ.

Solution

Method 1: Entrywise Computations

(a) The matrix $A^T B$ is $n \times n$ with entries

$$(A^T B)_{jj'} = \sum_{i=1}^m (A^T)_{ji} B_{ij'} = \sum_{i=1}^m a_{ij} b_{ij'}.$$

The trace is the sum of diagonal entries:

$$\operatorname{tr}(A^T B) = \sum_{j=1}^n (A^T B)_{jj} = \sum_{j=1}^n \sum_{i=1}^m a_{ij} b_{ij} = \sum_{i=1}^m \sum_{j=1}^n a_{ij} b_{ij}.$$

(b) From (a), we have

$$\operatorname{tr}(B^T A) = \sum_{i=1}^m \sum_{j=1}^n b_{ij} a_{ij} = \sum_{i=1}^m \sum_{j=1}^n a_{ij} b_{ij} = \operatorname{tr}(A^T B).$$

(c) Applying (a) with $B = A$,

$$\operatorname{tr}(A^T A) = \sum_{i=1}^m \sum_{j=1}^n a_{ij}^2 \geq 0.$$

Equality holds if and only if every term $a_{ij}^2 = 0$, i.e. all entries $a_{ij} = 0$. Hence $\operatorname{tr}(A^T A) = 0$ iff $A = 0$.

(d) The product BA^T is an $m \times m$ matrix with entries

$$(BA^T)_{ii'} = \sum_{j=1}^n B_{ij} (A^T)_{ji'} = \sum_{j=1}^n b_{ij} a_{i'j}.$$

Its trace is

$$\operatorname{tr}(BA^T) = \sum_{i=1}^m (BA^T)_{ii} = \sum_{i=1}^m \sum_{j=1}^n b_{ij} a_{ij} = \sum_{i=1}^m \sum_{j=1}^n a_{ij} b_{ij} = \operatorname{tr}(A^T B),$$

again coinciding with the expression in (a).

Method 2: Frobenius Inner Product and Cyclicity

Define the Frobenius inner product on $m \times n$ matrices by

$$\langle A, B\rangle_F := \operatorname{tr}(A^T B).$$

Then by definition,

$$\langle A, B\rangle_F = \sum_{i,j} a_{ij} b_{ij},$$

which proves (a).

Properties of the inner product immediately give:

- Symmetry: $\langle A, B\rangle_F = \langle B, A\rangle_F$ gives (b): $\operatorname{tr}(A^T B) = \operatorname{tr}(B^T A)$.
- Positive-definiteness: $\langle A, A\rangle_F = \operatorname{tr}(A^T A) \geq 0$ with equality only when $A = 0$, giving (c).

For (d), note that for conformable matrices,

$$\operatorname{tr}(XY) = \operatorname{tr}(YX),$$

whenever both products are square. Here $A^T B$ is $n \times n$ and BA^T is $m \times m$, so both traces are defined, and

$$\operatorname{tr}(A^T B) = \operatorname{tr}(BA^T).$$

This is just the cyclic property of trace applied to the pair A^T and B.

Problem 3

Problem

Let $1_n = (1, \dots, 1)^T \in \mathbb{R}^n$ and M_n be an $n \times n$ magic matrix whose rows, columns, and diagonals all sum to the same constant. Using only matrix operations on M_n, express all of its defining conditions without referencing individual entries.

Solution

Method 1: Row/Column/Diagonal Conditions in Matrix Form

Let the common sum be $s \in \mathbb{R}$. The defining conditions can be expressed as:

- **Row sums equal s:**
$$M_n\, 1_n = s\, 1_n.$$
- **Column sums equal s:**
$$M_n^T\, 1_n = s\, 1_n.$$
- **Main diagonal sums to s:**
$$\operatorname{tr}(M_n) = s.$$
- **Other diagonal sums to s:** Let J be the $n \times n$ permutation matrix with ones on the anti-diagonal and zeros elsewhere: $(J)_{i,n+1-i} = 1$. Then
$$\operatorname{tr}(JM_n) = s$$
encodes that the anti-diagonal entries of M_n sum to s.

Together, these matrix equalities express all four magic conditions (rows, columns, both diagonals) purely in terms of M_n and fixed matrices/vectors.

If in addition M_n is a *normal* magic matrix with entries $1, 2, \dots, n^2$, then the total sum of entries is $\frac{n^2(n^2+1)}{2}$, and since each of the n rows sums to s, we must have
$$ns = \frac{n^2(n^2+1)}{2} \quad \Rightarrow \quad s = \frac{1}{2}n(n^2+1).$$
In that case the conditions can be written explicitly as
$$M_n 1_n = \tfrac{1}{2}n(n^2{+}1)\, 1_n, \quad M_n^T 1_n = \tfrac{1}{2}n(n^2{+}1)\, 1_n, \quad \operatorname{tr}(M_n) = \tfrac{1}{2}n(n^2{+}1), \quad \operatorname{tr}(JM_n) = \tfrac{1}{2}n(n^2{+}1).$$

Method 2: Eigenvalue and Trace Interpretation

The vector 1_n is the (right) eigenvector of M_n associated with the eigenvalue s:
$$M_n 1_n = s 1_n,$$
and also a (left) eigenvector, since
$$1_n^T M_n = (M_n^T 1_n)^T = (s 1_n)^T = s 1_n^T.$$
Thus the magic property can be rephrased as:

- 1_n lies in both the right and left eigenspaces corresponding to eigenvalue s.
- The main diagonal sum equals the same eigenvalue: $\text{tr}(M_n) = s$.
- The anti-diagonal sum is also s, captured by $\text{tr}(JM_n) = s$.

All these statements use only matrix operations (multiplication, transpose, trace) and fixed vectors/matrices.

Problem 4

Problem

Given an $N \times n$ matrix $S = (s_{ij})$ where

$$s_{ij} = \begin{cases} 1 & \text{if patient } i \text{ exhibits symptom } j, \\ 0 & \text{otherwise,} \end{cases}$$

interpret in words, giving dimensions and entry descriptions:

(a) $S\,1_n$,

(b) $S^T\,1_N$,

(c) $S^T S$,

(d) SS^T.

Solution

Method 1: Entrywise Counting Interpretation

(a) $S1_n$ is an N-dimensional column vector:

$$S1_n \in \mathbb{R}^N.$$

Its i-th entry is

$$(S1_n)_i = \sum_{j=1}^{n} s_{ij},$$

the number of symptoms exhibited by patient i.

(b) $S^T 1_N$ is an n-dimensional column vector:

$$S^T 1_N \in \mathbb{R}^n.$$

Its j-th entry is

$$(S^T 1_N)_j = \sum_{i=1}^{N} s_{ij},$$

the number of patients who exhibit symptom j.

(c) $S^T S$ is an $n \times n$ matrix. Its (i, j)-th entry is

$$(S^T S)_{ij} = \sum_{k=1}^{N} S^T_{ik} S_{kj} = \sum_{k=1}^{N} s_{ki} s_{kj}.$$

Since $s_{ki}s_{kj} = 1$ exactly when patient k has both symptoms i and j, this counts the number of patients who simultaneously exhibit symptoms i and j. The diagonal entries $(S^T S)_{ii}$ are simply the counts of patients having symptom i.

(d) SS^T is an $N \times N$ matrix. Its (i,k)-th entry is

$$(SS^T)_{ik} = \sum_{j=1}^{n} s_{ij} s_{kj},$$

the number of symptoms shared by patient i and patient k. The diagonal entries $(SS^T)_{ii} = \sum_j s_{ij}$ give the number of symptoms of patient i.

Method 2: Gram Matrix Viewpoint

View each row of S as a vector $r_i \in \{0,1\}^n$ encoding the symptom profile of patient i, and each column as a vector $c_j \in \{0,1\}^N$ encoding which patients have symptom j.

Then:

- $S 1_n$ collects $\|r_i\|_1$ (the number of ones in each row), i.e. symptom counts per patient.
- $S^T 1_N$ collects $\|c_j\|_1$ (the number of ones in each column), i.e. patient counts per symptom.
- $S^T S$ is the Gram matrix of the column vectors c_j under the standard inner product:

$$(S^T S)_{ij} = c_i^T c_j,$$

  the number of patients sharing symptoms i and j.
- SS^T is the Gram matrix of the row vectors r_i:

$$(SS^T)_{ik} = r_i^T r_k,$$

  the number of symptoms shared by patients i and k.

Thus $S^T S$ captures symptom co-occurrence, while SS^T captures similarity between patients.

Problem 5

Problem

Let

$$C = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

(a) Describe how Cx acts on any $x \in \mathbb{R}^5$.

(b) Compute C^5.

Solution

Method 1: Action on Coordinates and Powers

Let $x = (x_1, \dots, x_5)^T$. Then

$$Cx = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} x_5 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}.$$

(a) Thus C performs a circular shift of the entries of x *down* by one position with wrap-around: the last component x_5 moves to the top, and each other component moves down one slot.

(b) Applying C repeatedly:

$$C^2x = C(Cx) = C(x_5, x_1, x_2, x_3, x_4)^T = (x_4, x_5, x_1, x_2, x_3)^T,$$

and in general each application shifts by one more position. After 5 applications, each entry has been shifted through all positions and returns to its original spot:

$$C^5x = x \quad \text{for all } x \in \mathbb{R}^5.$$

Therefore

$$C^5 = I_5,$$

the 5×5 identity matrix.

Method 2: Permutation Matrix and Cycle Structure

The matrix C is a permutation matrix corresponding to the cycle

$$(1\ 2\ 3\ 4\ 5)^{-1} = (1\ 5\ 4\ 3\ 2),$$

depending on the chosen convention. Its action on the standard basis vectors e_i is

$$Ce_1 = e_2, \quad Ce_2 = e_3, \quad Ce_3 = e_4, \quad Ce_4 = e_5, \quad Ce_5 = e_1,$$

or equivalently C represents a 5-cycle. A permutation matrix has order equal to the least common multiple of the lengths of its cycles; here the single cycle has length 5, so $C^5 = I_5$ and no smaller positive power equals the identity. This again implies that $C^5x = x$ for all x.

Problem 6

Problem

Represent a monochrome 3×3 image as a vector $x \in \mathbb{R}^9$ via row-major ordering. For each transformation $y = f(x)$ below, find the 9×9 matrix A such that $y = Ax$:

(i) Flip the image upside down.

(ii) Rotate the image clockwise by $90°$.

(iii) Translate up by 1 pixel and right by 1 pixel, filling new pixels with 0.

(iv) Set each pixel y_i to the average of its neighbors in row and column directions of the original.

Solution

Preliminaries: Indexing Convention

With row-major ordering, we identify

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \end{bmatrix} \quad \leftrightarrow \quad \begin{bmatrix} x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 \\ x_7 & x_8 & x_9 \end{bmatrix}.$$

All formulas below use this convention.

Method 1: Constructing A via Coordinate Mapping

(i) **Flip upside down.**

Flipping vertically swaps the first and third rows of the image. Thus the output pixels y_i satisfy

$$\begin{aligned} y_1 &= x_7, \quad y_2 = x_8, \quad y_3 = x_9, \\ y_4 &= x_4, \quad y_5 = x_5, \quad y_6 = x_6, \\ y_7 &= x_1, \quad y_8 = x_2, \quad y_9 = x_3. \end{aligned}$$

The corresponding matrix A_{flip} has one 1 in each row and column (a permutation matrix):

$$A_{\text{flip}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad y = A_{\text{flip}} x.$$

(ii) **Rotate clockwise by 90°.**

Writing the original as $\begin{bmatrix} x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 \\ x_7 & x_8 & x_9 \end{bmatrix}$, a 90° clockwise rotation yields

$$\begin{bmatrix} x_7 & x_4 & x_1 \\ x_8 & x_5 & x_2 \\ x_9 & x_6 & x_3 \end{bmatrix},$$

so

$$\begin{aligned} y_1 &= x_7, & y_2 &= x_4, & y_3 &= x_1, \\ y_4 &= x_8, & y_5 &= x_5, & y_6 &= x_2, \\ y_7 &= x_9, & y_8 &= x_6, & y_9 &= x_3. \end{aligned}$$

Thus

$$A_{\text{rot}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad y = A_{\text{rot}} x.$$

(iii) **Translate up by 1 pixel and right by 1 pixel, filling new pixels with 0.**

The transformation sends a pixel from position (r, c) (row, column) to position $(r-1, c+1)$ when this lies inside the 3×3 grid; otherwise it is discarded and the resulting pixel is set to zero. Valid original positions are:

$$(2,1) \mapsto (1,2), \quad (2,2) \mapsto (1,3), \quad (3,1) \mapsto (2,2), \quad (3,2) \mapsto (2,3).$$

In terms of indices (row-major):

$$\begin{aligned} (2,1) &\leftrightarrow x_4, & (1,2) &\leftrightarrow y_2, \\ (2,2) &\leftrightarrow x_5, & (1,3) &\leftrightarrow y_3, \\ (3,1) &\leftrightarrow x_7, & (2,2) &\leftrightarrow y_5, \\ (3,2) &\leftrightarrow x_8, & (2,3) &\leftrightarrow y_6. \end{aligned}$$

All other y_i are zero. Hence

$$A_{\text{trans}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad y = A_{\text{trans}}x.$$

(iv) **Average of neighbors in row/column directions.**

For each pixel at position (r, c), we take the average of its neighbors in the four directions: up, down, left, right (using only those that lie within the image). For example, the top-left pixel $(1, 1)$ has neighbors $(1, 2)$ and $(2, 1)$.

Writing the neighbor relationships in terms of the x_i-indices, we obtain:

$$\begin{aligned} y_1 &= \frac{1}{2}(x_2 + x_4), \\ y_2 &= \frac{1}{3}(x_1 + x_3 + x_5), \\ y_3 &= \frac{1}{2}(x_2 + x_6), \\ y_4 &= \frac{1}{3}(x_1 + x_5 + x_7), \\ y_5 &= \frac{1}{4}(x_2 + x_4 + x_6 + x_8), \\ y_6 &= \frac{1}{3}(x_3 + x_5 + x_9), \\ y_7 &= \frac{1}{2}(x_4 + x_8), \\ y_8 &= \frac{1}{3}(x_5 + x_7 + x_9), \\ y_9 &= \frac{1}{2}(x_6 + x_8). \end{aligned}$$

Thus the averaging matrix A_{avg} is

$$A_{\text{avg}} = \begin{bmatrix} 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{3} & 0 & \frac{1}{3} & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{3} & 0 & 0 & 0 & \frac{1}{3} & 0 & \frac{1}{3} & 0 & 0 \\ 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 \\ 0 & 0 & \frac{1}{3} & 0 & \frac{1}{3} & 0 & 0 & 0 & \frac{1}{3} \\ 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & \frac{1}{3} & 0 & \frac{1}{3} \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \end{bmatrix}, \quad y = A_{\text{avg}} x.$$

Method 2: Basis-Vector Construction (Conceptual)

Each of the above matrices can also be constructed systematically by specifying the image of the standard basis vectors $e_1, \dots, e_9$ under the transformation and stacking these images as columns:

$$Ae_j = \text{output when the original image has a single 1 at pixel } j \text{ and 0 elsewhere.}$$

For geometric transformations (flip, rotation, translation), each e_j maps to another basis vector or to 0, producing permutation-type columns. For the averaging operator, each e_j spreads to its neighbors with equal weights, giving columns with a few non-zero entries whose pattern mirrors the neighborhood structure. This procedure automatically yields the same matrices $A_{\text{flip}}, A_{\text{rot}}, A_{\text{trans}}, A_{\text{avg}}$ as above.