HOMEWORK # 2

Ahmad F. Taha

September 9, 2026

Due date of this homework: September 20, 2026, midnight.

This homework covers the mathematical fundamentals in Module 2. For computational problems, you may use **either MATLAB or Python**; you do not need to use both. The files in the `data/` folder work with both MATLAB and Python, and the `starter_code/` folder contains optional loading/solver templates to help you get started with the homework. Your solutions should all be typed on LaTeX. This is a very long homework, so please start early.

1. **Functions, sets, and basic function types.** Consider

$$f(x_1, x_2) = \sqrt{9 - x_1^2 - x_2^2}.$$

   (a) Find $\mathrm{dom}(f)$ and $\mathrm{range}(f)$.

   (b) Let $S = \{(x_1, 0) \mid -2 \le x_1 \le 2\}$. Find the image $f(S)$.

   (c) Let $T = [2, 3]$. Find the inverse image $f^{-1}(T)$ and describe it geometrically.

   (d) Classify each function below as linear, affine, quadratic, or none of these:

$$g(x) = 3x_1 - 2x_2, \qquad h(x) = 3x_1 - 2x_2 + 5,$$

$$q(x) = \frac{1}{2} x^T \begin{bmatrix} 4 & 1 \\ 1 & 2 \end{bmatrix} x + \begin{bmatrix} -2 \\ 3 \end{bmatrix}^T x + 1.$$

2. **Gradient, Hessian, and Taylor approximation.** For

$$f(x_1, x_2) = x_1^2 + 2x_1x_2 + 3x_2^2 - 4x_1 + 2x_2 + 7,$$

   let $x_0 = [1 \ \ -1]^T$ defines an operating point.

   (a) Compute $\nabla f(x)$ and $\nabla^2 f(x)$.

   (b) Evaluate $f(x_0)$, $\nabla f(x_0)$, and $\nabla^2 f(x_0)$.

   (c) Write the first-order Taylor approximation $f_{\text{lin}}(x)$ around x_0.

   (d) Write the second-order Taylor approximation $f_{\text{quad}}(x)$ around x_0.

   (e) Evaluate f, f_{lin}, and f_{quad} at $x = [1.1 \ \ -0.8]^T$. Explain why one approximation is exact here.

3. **Vector norms, distance, and inequalities.** Let

$$x = \begin{bmatrix} -2 \\ 1 \\ 3 \end{bmatrix}, \qquad y = \begin{bmatrix} 1 \\ -4 \\ 2 \end{bmatrix}.$$

   Recall the following inequalities, which we did not thoroughly go through in Module 2 (because I was blabbering about something else):

$$|x^T y| \le \|x\|_2 \, \|y\|_2 \qquad \text{(Cauchy–Schwarz inequality)}$$

   and, if $\frac{1}{p} + \frac{1}{q} = 1$ with $p \ge 1$,

$$\sum_{i=1}^{n} |x_i y_i| \le \|x\|_p \, \|y\|_q \qquad \text{(Hölder's inequality).}$$

   (a) Compute $\|x\|_1$, $\|x\|_2$, $\|x\|_\infty$ and the same three norms for y.

   (b) Compute $d_1(x, y)$, $d_2(x, y)$, and $d_\infty(x, y)$.

(c) Verify the Cauchy-Schwarz inequality numerically for x and y.

(d) Verify Hölder's inequality for $p = 1$ and $q = \infty$.

(e) Consider $N(z) = \sqrt{z_1^2 + 4z_2^2}$. Confirm that $N(\cdot)$ is a vector norm by relating it to a Euclidean norm after a linear scaling. Is $G(z) = \|z\|_2^2$ a norm? State the norm property that fails if your answer is no.

4. **Matrix norms, singular values, and norm bounds.** For any real (square or rectangular) matrix A, the *singular values* of A are defined as

$$\sigma_i(A) = \sqrt{\lambda_i(A^T A)},$$

where $\lambda_i(A^T A)$ are the eigenvalues of the symmetric matrix $A^T A$. These singular values are always real and nonnegative. They basically extend the notion of eigenvalues to rectangular matrices and are very useful for image/video processing. The nuclear norm (I did not discuss this in class) is defined by

$$\|A\|_* = \sum_i \sigma_i(A).$$

It is the sum of all singular values of A basically. Let

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

(a) Compute $\|A\|_1$, $\|A\|_\infty$, $\|A\|_F$, and $\|A\|_2$. For the 2-norm, use the eigenvalues of $A^T A$.

(b) Compute the singular values of A from the eigenvalues of $A^T A$, then compute the nuclear norm $\|A\|_*$.

(c) Find $\mathrm{rank}(A)$ and verify

$$\|A\|_2 \le \|A\|_F \le \sqrt{\mathrm{rank}(A)}\, \|A\|_2.$$

(d) Compute AB and verify numerically that $\|AB\|_F \le \|A\|_F \|B\|_F$.

(e) In two or three sentences, explain why the Frobenius norm is a valid matrix norm by viewing the entries of a matrix as one long vector.

5. **Eigenvalues, eigenvectors, inverse, and diagonalization.** Consider

$$A = \begin{bmatrix} 4 & 1 \\ 2 & 3 \end{bmatrix}.$$

For parts (a)–(d), **do not use** MATLAB's `eig` or Python's `numpy.linalg.eig`; show the manual calculation.

(a) Compute $\det(A - \lambda I)$ and find all eigenvalues.

(b) Find one eigenvector associated with each eigenvalue.

(c) Compute A^{-1} manually and verify $AA^{-1} = I$.

(d) Construct T and D such that $A = TDT^{-1}$.

(e) Now verify your eigenvalues, eigenvectors, and diagonalization using MATLAB or Python.

6. **Square-image reconstruction from eigenanalysis of $A^T A$.** Use `data/square_infrastructure_image.png`. The image is a 128×128 grayscale image, so it can be represented by a square matrix A. A numeric copy is also provided as `square_infrastructure_image_matrix.csv`.

You have not covered singular values formally in class. For this problem, use the following self-contained construction:

- Form $C = A^T A$.
- Compute eigenpairs (λ_i, v_i) of C, ordered so that $\lambda_1 \geq \lambda_2 \geq \cdots \geq 0$.
- Define the singular values of A by $\sigma_i = \sqrt{\lambda_i}$.
- For each nonzero σ_i, define
$$u_i = \frac{Av_i}{\sigma_i}.$$
- For a chosen k, collect the first k vectors into
$$V_k = [v_1 \ \cdots \ v_k], \qquad U_k = [u_1 \ \cdots \ u_k], \qquad \Sigma_k = \operatorname{diag}(\sigma_1, \dots, \sigma_k),$$
and then define the rank-k approximation
$$A_k = U_k \Sigma_k V_k^T.$$

This approximation tells you that any video or image can be compressed into a rank-k approximation without needing to include all singular values that are too tiny to contribute to the image or video. Do the following using MATLAB or Python.

(a) Read the image, convert it to a matrix with entries in $[0, 1]$, and report its dimensions and numerical rank.

(b) Form $C = A^T A$. Compute the eigenvalues and eigenvectors of C, and sort the eigenvalues from largest to smallest. Explain why these eigenvalues should be nonnegative, up to roundoff.

(c) Recover singular values using $\sigma_i = \sqrt{\lambda_i}$. Plot the first 40 singular values on a semilogarithmic scale.

(d) Using the recipe above, form A_k for $k = 1, 5, 10, 20, 40$ and display the reconstructed images.

(e) For each k, compute the relative reconstruction error
$$\frac{\|A - A_k\|_F}{\|A\|_F}.$$
Which k gives a visually reasonable reconstruction?

(f) In plain language, explain what the largest eigenvalues of $A^T A$ mean for the image and why rapid eigenvalue decay makes image compression possible.

7. **Rank, determinant, and null space.** Consider
$$M = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 1 & 0 \end{bmatrix}.$$

(a) Reduce M to row-echelon form and compute $\operatorname{rank}(M)$.

(b) Find a basis for $\operatorname{Null}(M)$.

(c) Compute $\det(M)$.

(d) Explain the relationship among your answers to (a)–(c), invertibility, and the existence of a zero eigenvalue.

8. **Symmetry and definiteness.** Classify each symmetric matrix below as positive definite, positive semidefinite, negative definite, negative semidefinite, or indefinite. Use principal minors and/or eigenvalues, and show your work.

$$A_1 = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \quad A_3 = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}.$$

Then consider

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

All leading principal minors of C are nonnegative. Show that C is nevertheless *not* positive semidefinite by finding a principal minor that is negative. Briefly explain why leading principal minors alone are insufficient for testing positive semidefiniteness.

9. **Low-rank structure in noisy time-series data.** Use `data/time_series_noisy.csv`. Column 1 contains time in minutes; columns 2–17 contain 16 infrastructure sensor signals sampled every 10 minutes for five days.

For any matrix X, define its singular values as the square roots of the eigenvalues of $X^T X$, ordered from largest to smallest.

(a) Plot any four sensor signals versus time.

(b) Form a matrix $X \in \mathbb{R}^{16\times 720}$ with one sensor per row, and subtract the temporal mean from each row. Compute the default numerical rank of the centered matrix.

(c) Filter each sensor with an 11-sample moving average (about 110 minutes), form the centered filtered matrix X_f, and compute its default numerical rank.

(d) Compute the singular values of both centered matrices and plot σ_i/σ_1 for the raw and filtered data on the same axes.

(e) Define the *99% effective rank* as the smallest r satisfying

$$\frac{\sum_{i=1}^{r} \sigma_i^2}{\sum_i \sigma_i^2} \geq 0.99.$$

Report this effective rank before and after filtering.

(f) Form the rank-r approximation of X_f using the 99% effective rank and compute the relative Frobenius error. Explain what this example shows about low-rank structure in correlated time-series data after noise is reduced.

10. **Moore-Penrose inverse for rectangular systems.**

(a) Let

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}, \qquad b = \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix}.$$

Verify that A has linearly independent columns, compute

$$A^{\dagger} = (A^T A)^{-1} A^T,$$

and then compute $x = A^{\dagger} b$. Verify that $Ax = b$.

(b) Let

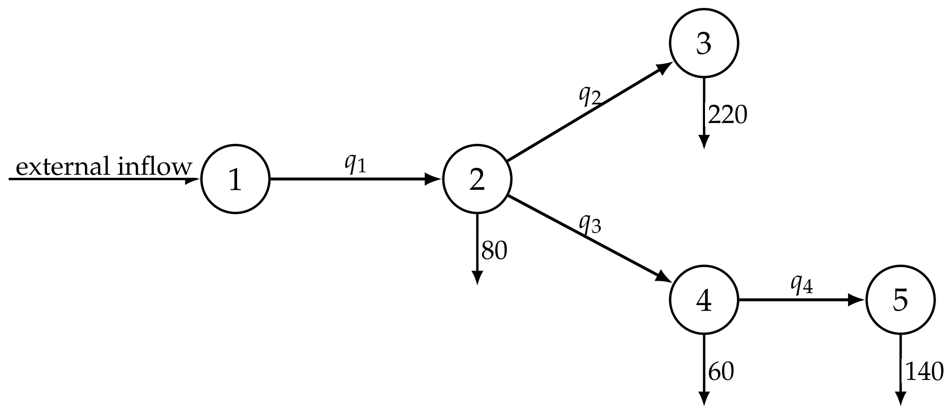
$$G = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}, \qquad c = \begin{bmatrix} 3 \\ 1 \end{bmatrix}.$$

Verify that G has linearly independent rows, compute

$$G^\dagger = G^T(GG^T)^{-1},$$

and compute one solution $z = G^\dagger c$. Verify that $Gz = c$. Why is the solution not unique?

11. **A small traffic network: write and solve $Aq = b$ manually.** The arrows $q_1, \ldots, q_4$ are unknown link flows in vehicles/hour. The downward arrows are known departures from the network. Assume flow conservation at nodes 2–5.



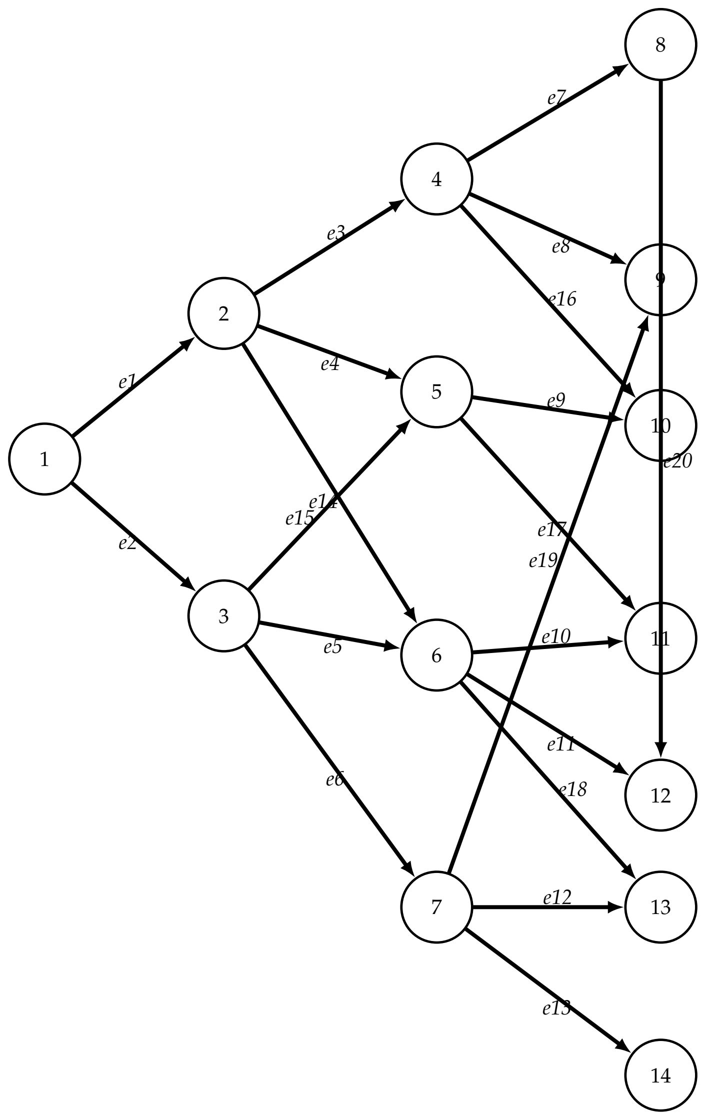
(a) Write the four conservation equations in the unknown vector $q = [q_1\ q_2\ q_3\ q_4]^T$.

(b) Put the equations in matrix form $Aq = b$.

(c) Solve the system manually and report all four link flows.

(d) Compute $\det(A)$ and $\operatorname{rank}(A)$ and explain why the solution is unique.

12. **A larger traffic network: incidence matrices and coding.** The larger directed network below has 14 nodes and 20 links. Sorry the figure is ugly and arrows are overlapping. Use the files `traffic_incidence.csv`, `traffic_injections.csv`, `traffic_edges.csv`, and `traffic_measurements.csv`.



The incidence convention is: each edge column has $+1$ at its tail node and -1 at its head node. Thus the conservation model is

$$Bq = s,$$

where positive entries of s represent net external inflow and negative entries represent net withdrawal.

(a) Load B and s. Report their dimensions, $\operatorname{rank}(B)$, and $\sum_i s_i$. Explain why one nodal balance equation is redundant.

(b) Remove row 1 from B and s to obtain $B_r q = s_r$. Report $\operatorname{rank}(B_r)$. Is the system uniquely solvable from conservation alone? Explain using the dimensions and rank.

(c) Because B_r has independent rows, compute one solution using the right pseudoinverse from Module 2:

$$q_0 = B_r^T (B_r B_r^T)^{-1} s_r.$$

Report $\|B_r q_0 - s_r\|_2$.

(d) The file `traffic_measurements.csv` contains measured flows on seven links. Construct a selector matrix H so that $Hq = m$. Stack conservation and measurements:

$$\underbrace{\begin{bmatrix} B_r \\ H \end{bmatrix}}_{A_{\text{aug}}} q = \underbrace{\begin{bmatrix} s_r \\ m \end{bmatrix}}_{b_{\text{aug}}}.$$

Report the dimensions and rank of A_{aug}, solve for all 20 link flows, and verify the residual $\|A_{\text{aug}} q - b_{\text{aug}}\|_2$.

(e) In two or three sentences, explain what the seven flow measurements add that conservation equations alone cannot determine in a meshed traffic network.

13. **Writing a multi-input, multi-output infrastructure model in state-space form.** A linearized four-storage water system is modeled by

$$\dot{x}_1 = -0.30x_1 + 0.10x_2 + 0.04x_3 + 0.50u_1,$$

$$\dot{x}_2 = 0.08x_1 - 0.25x_2 + 0.07x_4 + 0.20u_1,$$
$$\dot{x}_3 = 0.05x_1 - 0.20x_3 + 0.06x_4 + 0.40u_2,$$
$$\dot{x}_4 = 0.10x_2 + 0.08x_3 - 0.22x_4 + 0.15u_2.$$

The measured outputs are

$$y_1 = x_1, \qquad y_2 = x_2 + x_4, \qquad y_3 = x_3 + 0.05u_2.$$

(a) Define $x(t)$, $u(t)$, and $y(t)$ as vectors.

(b) Write the model as

$$\dot{x}(t) = Ax(t) + Bu(t), \qquad y(t) = Cx(t) + Du(t).$$

Give all entries of A, B, C, and D.

(c) State the dimensions of A, B, C, and D and verify they agree with $n = 4$ states, $m = 2$ inputs, and $p = 3$ outputs.

14. **Numerically solving the state-space model with `ode45` or `solve_ivp`.** Use the state-space model from Problem 13 with

$$x(0) = \begin{bmatrix} 2.0 \\ 1.5 \\ 1.0 \\ 0.5 \end{bmatrix}.$$

For $0 \le t \le 30$, use the piecewise-constant inputs

$$u_1(t) = \begin{cases} 1.0, & t < 12, \\ 0.4, & t \ge 12, \end{cases} \qquad u_2(t) = \begin{cases} 0.2, & t < 5, \\ 1.0, & t \ge 5. \end{cases}$$

Use MATLAB's `ode45` or Python's `scipy.integrate.solve_ivp`.

(a) Implement a function that returns $\dot{x} = Ax + Bu(t)$.

(b) Simulate the system on $0 \le t \le 30$ and plot all four states versus time.

(c) Compute $y(t) = Cx(t) + Du(t)$ from the simulated state trajectory and plot all three outputs.

(d) Report $x(30)$ and $y(30)$.

(e) Briefly explain the roles of the state vector, input vector, and output vector in this model. Your answer should be conceptual, not a description of MATLAB/Python syntax.