Computational Matrix Methods Project

Elliot Porter

October 2025

---

1. Introduction

This document collects a set of computationally oriented matrix-methods problems that extend the geometric and theoretical intuition from EE263 into numerical and algorithmic practice. The problems in `problems.tex` focus on Gaussian elimination, QR factorization, conditioning, SVD, and recursive least-squares.

---

2. Computational Problem Set: Numerical Matrix Methods

This optional problem set is designed to develop practical computational intuition for matrix methods, building on the geometric material of EE263 and extending toward the algorithmic perspective emphasized by Gerritsen's *Introduction to Matrix Computations*.

Problem 1: Gaussian Elimination and Pivoting

Goal: Understand how elimination behaves numerically.

(a) Implement Gaussian elimination on $A \in \mathbb{R}^{n \times n}$ with and without partial pivoting.

(b) Test both methods on an ill-conditioned matrix, e.g. the Hilbert matrix $H_{ij} = 1/(i + j - 1)$ for $n = 8$.

(c) Compare the solutions $\hat{x}$ with the true solution x_{true} using the relative error

$$\text{err} = \frac{\|\hat{x} - x_{\text{true}}\|_2}{\|x_{\text{true}}\|_2}.$$

(d) Compute the condition number $\kappa(A) = \|A\|_2 \|A^{-1}\|_2$ and discuss its relationship to the observed errors.

Expected outcome: Demonstrate that pivoting dramatically improves numerical stability.

Problem 2: QR Factorization and Least Squares

Goal: Show why QR is preferred to normal equations for solving least squares.

(a) For a tall random matrix $A \in \mathbb{R}^{m \times n}$ with $m > n$, solve

$$\min_x \|Ax - y\|_2$$

by

i. the normal equations $(A^T A)x = A^T y$, and

ii. QR factorization $A = QR$, $x = R^{-1} Q^T y$.

(b) Compare the residual $\|Ax - y\|_2$ and the runtime for both methods.

(c) Gradually scale one column of A by 10^k ($k = 0, \ldots, 6$) and plot the error versus $\kappa(A)$.

Expected outcome: Observe that QR maintains accuracy as $\kappa(A)$ grows, while normal equations degrade.

Problem 3: Conditioning and Sensitivity

Goal: Quantify how perturbations in A and b affect the computed solution.

(a) Choose a random nonsingular A and compute $\kappa(A)$ using the 2-norm.

(b) Perturb A and b slightly: $A' = A + \Delta A$, $b' = b + \Delta b$.

(c) Solve both systems and compute relative changes

$$\frac{\|\Delta x\|}{\|x\|}, \qquad \frac{\|\Delta A\|}{\|A\|}, \qquad \frac{\|\Delta b\|}{\|b\|}.$$

(d) Verify empirically that

$$\frac{\|\Delta x\|}{\|x\|} \lesssim \kappa(A)\left(\frac{\|\Delta A\|}{\|A\|} + \frac{\|\Delta b\|}{\|b\|}\right).$$

Expected outcome: Relate the condition number directly to error amplification.

Problem 4: Singular Value Decomposition and Pseudo-Inverse

Goal: Compute stable solutions for nearly rank-deficient systems.

(a) Construct $A \in \mathbb{R}^{6\times 4}$ with two almost-collinear columns.

(b) Compute the SVD $A = U\Sigma V^T$.

(c) Form the Moore–Penrose pseudo-inverse $A^+ = V\Sigma^+U^T$.

(d) Solve $x = A^+y$ for several y and compare to x from least squares via QR.

(e) Set small singular values to zero manually to explore regularization.

Expected outcome: Show how SVD reveals numerical rank and why A^+ provides the minimum-norm LS solution.

Problem 5: Regularized and Recursive Least Squares

Goal: Explore stabilization and efficient updates.

(a) Regularized least squares (ridge regression). Solve

$$x_\lambda = (A^TA + \lambda I)^{-1}A^Ty,$$

for $\lambda \in [10^{-6}, 10^2]$ and plot $\|Ax_\lambda - y\|_2$ versus $\|x_\lambda\|_2$. Discuss the trade-off between accuracy and regularization.

(b) Recursive least squares. Given rows a_i^T of A arriving sequentially, update

$$P_{k+1}^{-1} = P_k^{-1} - \frac{P_k^{-1} a_{k+1} a_{k+1}^T P_k^{-1}}{1 + a_{k+1}^T P_k^{-1} a_{k+1}}.$$

Show that P_{k+1}^{-1} equals $(A_{1:k+1}^T A_{1:k+1})^{-1}$ without recomputing from scratch.

Expected outcome: Recognize how regularization and recursion improve numerical stability and efficiency.

Recommended order:

1. Gaussian elimination (stability through pivoting)
2. QR factorization (stable least-squares)
3. Conditioning (error amplification)
4. SVD and pseudo-inverse (rank and regularization)
5. Regularized / recursive least-squares (stability + efficiency)