University of Toronto
Department of Electrical and Computer Engineering

ECE367 MATRIX ALGEBRA AND OPTIMIZATION

Problem Set #4
Autumn 2020

Prof. S. C. Draper **Due:** 5pm (Toronto time) Friday, 06 November 2020

Homework policy: Problem sets must be turned by the due date and time. Late problem sets will receive deductions for lateness. See the information sheet for futher details. The course text "Optimization Models" is abbreviated as "OptM". Also, see PS01 for details of the "Non-graded", "graded" and "optional" problem categories.

Note: In the categorization below Graded problems are highlighted in **red boldface** **Problem Set #4 problem categories:** A quick categorization by topic of the problems in this problem set is as follows:

- Least squares: Problems 4.1, 4.2, **4.4**
- Applications of least squares: Problems 4.3, **4.5**, **4.6**

NON-GRADED PROBLEMS

Problem 4.1 (Least squares and total least squares)
OptM Problem 6.1, the part on finding the least-squares line. The part on total least-squares is optional.

Problem 4.2 (ℓ_2 regularized least squares), from a previous exam

This problem considers optimization problems of the form

$$\min_{x \in \mathbb{R}^2} \|Ax - y\|_2^2 + \gamma \|x\|_2^2 \tag{1}$$

where $\gamma \in \mathbb{R}$ and $\gamma \geq 0$. In this problem we denote the x that minimizes (1) as x_γ^*.

(a) First consider the case where

$$A = \begin{bmatrix} 1 & 1 \\ 1 & -1 \\ 2 & 1 \end{bmatrix}, \qquad y = \begin{bmatrix} 4 \\ -2 \\ 5 \end{bmatrix}, \qquad \gamma = 0.$$

Find x_0^*, the optimal variable that solves (1) for the parameters given. (To be doubly clear, $x_0^* = x_\gamma^*|_{\gamma=0}$, which is read as "$x_\gamma^*$ evaluated at $\gamma = 0$".)

(b) Re-derive the following relation derived in class, *fully justifying* your steps:

$$\|Ax - y\|_2^2 = \|Ax_0^* - y\|_2^2 + \|A(x - x_0^*)\|_2^2,$$

again note that $x_0^* = x_\gamma^*|_{\gamma=0}$.

Parts (c) and (d) of this problem concern Figure 1. In Figure 1 we plot two *possible* paths for x_γ^* as a function of γ for the same values of A and y that are specified in part (a).

(c) Make a sketch of Fig. 1 in your solutions. To your sketch add some level sets of the objective of (1) for the case where $\gamma = 0$. You must show below work that mathematically justifies your sketches of the level sets.

(d) *You must base your answer to this part on your answer to part (c)*:

Which path does x_γ^* follow as one increases γ? In the space provided below both clearly indicate *which* path (either the upper, dashed, "Path A" or the lower, solid, "Path B") you think x_γ^* follows and provide an explanation for your choice *based on your answer to part (c)*.

Problem 4.3 (Model for enzyme kinetics)
OptM Problem 6.6 parts 1 and 2. (Note that for some reason Problem 6.6 is not given a title in OptM.)

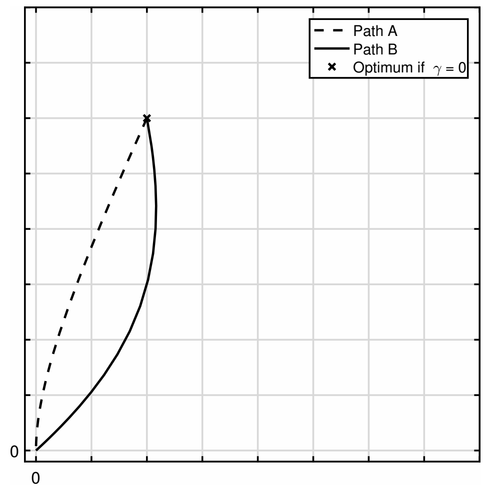


Figure 1: Path followed by x^*_γ as a function of γ.

GRADED PROBLEMS

Problem 4.4 (Solving least squares problems using the Moore-Penrose pseudoinverse) In this problem you derive the result that the Moore-Penrose pseudoinverse can be used to solve least squares problems for overdetermined, underdetermined, and rank-deficient systems. Recall that least square problems consider the system of linear equations $Ax = b$ where $A \in \mathbb{R}^{m,n}$ and $\text{rank}(A) = r$. If the compact SVD of A is $A = U_r \Sigma V_r^T$ where $\Sigma \in \mathbb{R}^{r,r}$ is a positive definite diagonal matrix of (non-zero) singular values, $U_r \in \mathbb{R}^{m,r}$ and $V_r \in \mathbb{R}^{n,r}$ each contain orthonormal columns, then the Moore-Penrose pseudoinverse is $A^\dagger = V_r \Sigma^{-1} U_r^T$.

(a) Recall that the *overdetermined* least squares problem considers an $A \in \mathbb{R}^{m,n}$ when $m > n$ (more constraints in the y vector than parameters in the x vector). Here the objective is to find an x that minimizes $\|Ax - y\|_2$. Show that an optimal solution is $x^* = A^\dagger y$. To show this recall that an optimal solution x^* must satisfy the "normal" equations $A^T A x^* = A^T y$. Verify that $x^* = A^\dagger y$ satisfies the normal equations.

(b) Recall that the *underdetermined* least squares problem considers an $A \in \mathbb{R}^{m,n}$ when $m < n$ (fewer constraints in the y vector than parameters in the x vector). Here the objective is to find the x that minimizes $\|x\|_2$ while satisfying $Ax = y$ (equivalently, satisfying $\|Ax - y\|_2 = 0$). Show that the optimal solution $x^* = A^\dagger y$. To show this recall that the optimal solution x^* must satisfy two conditions: (i) $x^* \in \mathcal{R}(A^T)$ and (ii) $Ax^* = y$. Verify that $x^* = A^\dagger y$ satisfies (i) all the time. Under what conditions does x^* satisfy condition (ii)? (Hint, think about the rank of A.)

In the above two parts we haven't explicitly considered the role of the rank r of the A matrix. But we also note that the only place where the rank of A comes into the discussion of parts (a) and (b) is in the discussion of condition (ii) of part (b).

Recall that $r \leq \min\{m, n\}$. When $r = \min\{m, n\}$ the A matrix is full column rank in the overdetermined problem and is full row rank in the underdetermined problem. In both these full-rank cases x^* has the simple expression presented in class. Now we consider what happens when $r < \min\{m, n\}$.

(c) In this part consider the situation where $\text{rank}(A) = r < m < n$. This is a "rank-deficient" underdetermined least squares problem. If we set $x^* = A^\dagger y$, what characteristics do Ax^* and $\|x^*\|_2$ satisfy? (Hint: this is a type of hybrid problem that at the same time can have characteristics of both overdetermined and underdetermined least squares.)

Problem 4.5 (Optimal control of a unit mass)

Consider a unit mass with position $x(t)$ and velocity $\dot{x}(t)$ subject to force $f(t)$, where the force is piecewise constant over intervals of duration one second, i.e., $f(t) = p_n$ for $n - 1 < t \leq n$, $n = 1, \cdots, 10$; we consider the system for 10 seconds in total. Ignore friction. Assume the mass has zero initial position and velocity, i.e., $x(0) = \dot{x}(0) = 0$.

(a) Derive the "state-space" equations that describe a discrete-time version of the dynamics of this system. In particular, derive relationships between $x(n)$ and $\dot{x}(n)$ in terms of $x(n-1)$, $\dot{x}(n-1)$, and the driving force p_n, for each $n \in \{1, 2, \ldots, 10\}$. (You will need to use your knowledge of basic physics to determine these relationships.) The two equations you derive should be

$$x(n) = x(n-1) + \dot{x}(n-1) + (1/2)p_n,$$
$$\dot{x}(n) = \dot{x}(n-1) + p_n,$$

which you can then stack up in vector form to form the state-space equations

$$\begin{bmatrix} x(n) \\ \dot{x}(n) \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}}_{A} \begin{bmatrix} x(n-1) \\ \dot{x}(n-1) \end{bmatrix} + \underbrace{\begin{bmatrix} 1/2 \\ 1 \end{bmatrix}}_{b} p_n. \tag{2}$$

We note that this continuous time system can only be discretized exactly because in this system the control effort $f(t)$ is held constant over each time interval. (At this point we recommend you remind yourself of the discussion of state-space models of linear dynamical systems in OptM Exercise 3.4.)

(b) Find the p_n, $n \in [10]$ that minimizes

$$\sum_{n=1}^{10} p_n^2$$

subject to the following specifications: $x(10) = 1$, $\dot{x}(10) = 0$. Plot the optimal f, the resulting x and $\dot{x}$. (I.e., plot $f(t)$, $x(t)$, and $\dot{x}(t)$ for $0 \leq t \leq 10$.) Give a short intuitive explanation of what you see.

(c) Suppose that we add one more specification $x(5) = 0$, i.e., we require the mass to be at position 0 at time 5. Plot the optimal f, the resulting x and $\dot{x}$. Give a short intuitive explanation of what you see.

Problem 4.6 (CAT scan imaging)

In this problem you will obtain a basic understanding of the math behind computer aided tomography (CAT) scanning. Before proceeding we recommend you read the Example 6.6 (*CAT scan imaging*) in OptM to obtain a basic understanding about the CAT scan. In the following parts, we assume the same meaning for y, A and x as in Example 6.6. Download the file `scanImage.p` from the course website. Although you will observe that the file is encrypted, this file defines a MATLAB function called `scanImage(M)` that takes in one optional argument and produces one output. You can execute this function just the same way you execute other MATLAB functions. This function encapsulates a "virtual CAT scanner" that is able to scan 2-d images. Consider a grayscale image of height h and width w. Such an image can be represented as a $h \times w$ matrix M with values between 0 and 255. Let $n = hw$ be the total number of pixels (in CAT-scan terminology this is the number of "voxels", as described in OptM). If you execute the function with M as the argument, it will return a vector $y \in \mathbb{R}^m$ which is the scan result. Similar to OptM Example 6.6, y is the vector of log-intensity ratios, and m corresponds to the number of beams used for the scan. Note that for the provided scanner, you have no control over the number of beams used or how beams are positioned across the image. However, the dimension m and the positioning of beams is a function only of h and w. This means that any two images which are of the same dimensions will be scanned with identical beam setups.

Similar to OptM, let us denote the vectorized image M by the n-dimensional column vector x. The input x can be related to the output y of the scanner by matrix $A \in \mathbb{R}^{m \times n}$ as $y = Ax$. In OptM Example 6.6, A is computed analytically, using geometric properties of the beam setup. For this numerical problem, even if you knew the positions (and angles) of the beams, computing A analytically would be … tedious. We propose an alternative approach to estimate A. Observe that

$$[y_1\, y_2\, \cdots\, y_m]^T = \left[a^{(1)}\, a^{(2)}\, \cdots\, a^{(n)}\right][x_1\, x_2\, \cdots\, x_n]^T, \tag{3}$$

where $a^{(i)}$ is the ith column of A. In the provided scanner, you have control over the input image. Consider the case where you obtain an image M_1 by setting the first pixel (the $(1, 1)$ coordinate of M) to 1 and all others to 0. The vectorized representation of M_1 will be $[1, 0, \cdots, 0]^T$. If you put M_1 through the scanner, the resulting y should be equal to $a^{(1)}$ as per (3). This way, by turning each pixel on while all others are off, you can estimate each of the columns in A. As you may guess, this is not how things work with real-world CAT scanners, but eases our development herein.

(a) Now, using the method described in the text, write a MATLAB script to estimate A when $h = 50$ and $w = 50$. In this case you will observe the dimension of the scanner output, i.e., m is 1950. Since A is an $m \times n$ matrix, we can visualize A by treating it as an image. Use the MATLAB function `imshow(A, [])` to display A as an image and include a scaled down version of the image with your answers. Note that if `imshow(A, [])` displays a mostly black image, `imshow(-A, [])` will display a mostly white image.

(b) Till this point, you used the provided scanner function `scanImage(·)` to get the outputs for known images, yielding an estimate for A. In this part, you are given the scanner output y of an unknown image M_{un}, which has dimensions $h = 50$ and $w = 50$. The scanner output y for the image M_{un} can be obtained by executing `y=scanImage`, without passing an input to the function. (We use an encrpyted function so that in this part you need to determine an estimate of the image.) Since y and A are known, you can solve for an x that satisfies $y = Ax$. You will find that $\operatorname{rank} A < m < n$ so this is an underdetermined and rank-deficient system of equations. However, the matrix A may also have non-zero singular values that are extremely close to zero which, to avoid numerical instabilities, you will need to assume to be zero. To choose an appropriate effective rank to work with, plot the r singular values σ_i of A versus their index $i \in [r]$. Visually inspect your plot to choose an effective minimum non-zero singular value. After solving for x, obtain the image M_{un} and include with your answers. What is the hidden message in the scanned image?

In a real-world CAT scanner, the number of voxels (n) usually is much larger than the number of beams (m) used to produce the vector y. Rather than least squares the 'inverse Radon transform' is typically used to solve for the vector x. Note that the number of voxels considered is proportional to the resolution of the image obtained. A typical CAT scanner provides around 0.5mm resolution, which means two adjacent voxels are around 0.5mm apart.

Problem 4.4

Part a

Show that optimal solution $x^* = A^+y$ solves the normal equations

$$A^TAx^* = A^Ty \Rightarrow A^TAA^+y = A^Ty$$
$$A^TAV_r\Sigma^{-1}U_r^Ty = A^Ty$$
$$U_r\Sigma V_r^TV_r\Sigma^{-1}U_r^Ty = y$$
$$U_r\Sigma I\Sigma^{-1}U_r^Ty = y$$
$$U_rIU_r^Ty = y$$
$$y = y, \text{ as required}$$

Part b

i) $x^* = A^+y$

Normal equation: $A^TAx^* = A^Ty$

$$(A^TA)(A^TA)^{-1}x^* = A^T(A^TA)^{-1}y$$
$$x^* = A^T(A^TA)^{-1}y$$

$(A^TA)^{-1}$ is some matrix, however, the above shows that x^* is indeed in the range of A^T.

ii) $Ax^* = y \Rightarrow AA^+y = y$

A^+ is a right inverse only if A is full rank. This means that $AA^+ = I$ only if $\text{rank}(A) = m$

$\therefore x^*$ satisfies the solution only if A is full rank

Part c

rank(A) = $r < m < n$, short fat matrix

If $x^* = A^+ y \Rightarrow Ax^* = AA^+ y$

In this case, the rows of matrix A are linearly dependent, which means that at least one row is a scalar multiple of another row.

$x^* = A^+ y = V_r \Sigma^{-1} U_r^T y \Rightarrow \|x^*\|_2 = \|\Sigma^{-1} U_r^T y\|_2$

$\geq |\Sigma^{-1} U_r^T y|$

$= \frac{|U_r^T y|}{\sigma_{min}}$, where σ_{min} is the smallest singular value of A

U_r in this case represents the last non-zero column of U in the SVD of A

$Ax^* = AA^+ y$, since A is not full rank in this case, the matrix AA^+ will also not be full rank. $\therefore$ Ax^* will be equal to a deficient rank matrix times y, which will lead to multiple solutions as expected by the original short, fat matrix.

Problem 4.5

Part a

Based on general physics: $\vec{F}_{net} = m\vec{a} = p_n \quad \forall n \in [10]$

$\frac{d^2x}{dt^2} = p_n$, where dt in this case is simply the change in n

$\Rightarrow v(n) - v(n-1) = p_n$

$v(n) = v(n-1) + p_n$ Also, $x(n) - x(n-1) = v(n-1) + 0.5p_n$

$\dot{x}(n) = \dot{x}(n-1) + p_n$ $x(n) = x(n-1) + \dot{x}(n-1) + 0.5p_n$

Putting this in a matrix: $\begin{pmatrix} x \\ \dot{x} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} x(n-1) \\ \dot{x}(n-1) \end{pmatrix} + \begin{pmatrix} 0.5 \\ 1 \end{pmatrix} p_n$

Part b

$\sum_{n=1}^{10} p_n^2 = \|p\|_2$, $\begin{pmatrix} x(10) \\ \dot{x}(10) \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} x(9) \\ \dot{x}(9) \end{pmatrix} + \begin{pmatrix} 0.5 \\ 1 \end{pmatrix} p_{10} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$

$\Rightarrow x(1) = 0.5p_1$ $x(2) = 0.5p_1 + p_1 + 0.5p_2$

$\dot{x}(1) = p_1$ $\dot{x}(2) = p_1 + p_2$

$x(3) = 2.5p_1 + 1.5p_2 + 0.5p_3$ $x(4) = 3.5p_1 + 2.5p_2 + 1.5p_3 + 0.5p_4$

$\dot{x}(3) = p_1 + p_2 + p_3$ $\dot{x}(4) = p_1 + p_2 + p_3 + p_4$

$\dots \begin{pmatrix} x(10) \\ \dot{x}(10) \end{pmatrix} = \begin{pmatrix} 9.5 & 8.5 & 7.5 & 6.5 & 5.5 & 4.5 & 3.5 & 2.5 & 1.5 & 0.5 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} p = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$

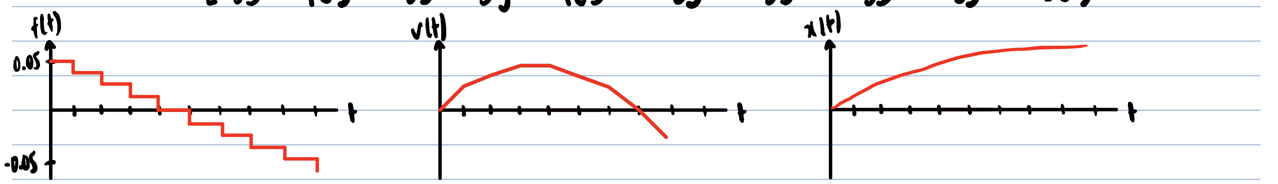
Let the above matrix be denoted as A. A in this case is full rank

$\Rightarrow p^* = A^T(AA^T)^{-1}\begin{pmatrix} 1 \\ 0 \end{pmatrix}$

$= A^T\begin{pmatrix} 332.5 & 50 \\ 50 & 10 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix}$

$= A^T\begin{pmatrix} 10 & -50 \\ -50 & 332.5 \end{pmatrix} \cdot \frac{1}{3325-2500}\begin{pmatrix} 1 \\ 0 \end{pmatrix}$

$= \left[\frac{3}{55} \quad \frac{7}{165} \quad \frac{1}{33} \quad \frac{1}{55} \quad \frac{1}{165} \quad -\frac{1}{65} \quad -\frac{1}{55} \quad -\frac{1}{33} \quad -\frac{7}{65} \quad -\frac{3}{55}\right)^T$



$f(t)$: piecewise constant graph
$v(t)$: piecewise linear graph
$x(t)$: piecewise quadratic graph

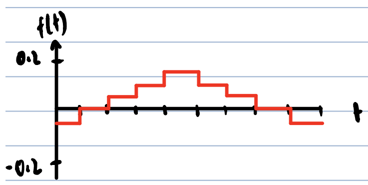
Part c

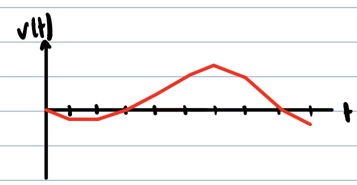
$$\begin{pmatrix} x(10) \\ \dot{x}(10) \\ x(5) \end{pmatrix} = \begin{pmatrix} 9.5 & 8.5 & 7.5 & 6.5 & 5.5 & 4.5 & 3.5 & 2.5 & 1.5 & 0.5 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 4.5 & 3.5 & 2.5 & 1.5 & 0.5 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} p = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

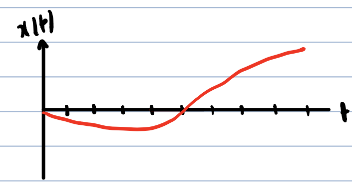
Matrix A is again full rank and can be solved the same way as part b.

$$p = A^T(AA^T)^{-1}\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

$$= \begin{pmatrix} -\frac{1}{22} & -\frac{1}{132} & \frac{1}{33} & \frac{3}{44} & \frac{7}{66} & \frac{31}{330} & \frac{7}{220} & -\frac{1}{33} & -\frac{61}{660} & \frac{17}{110} \end{pmatrix}^T$$



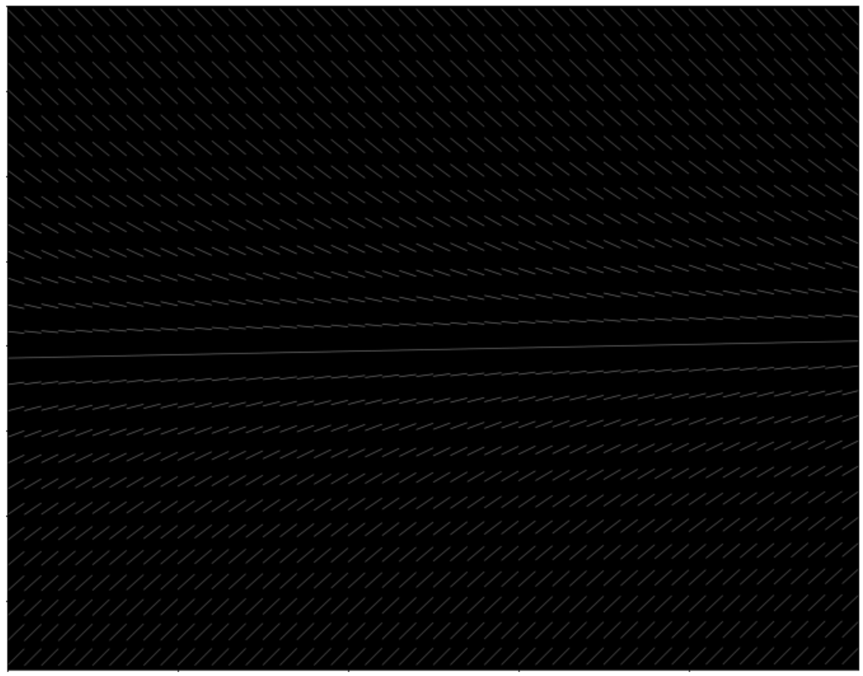




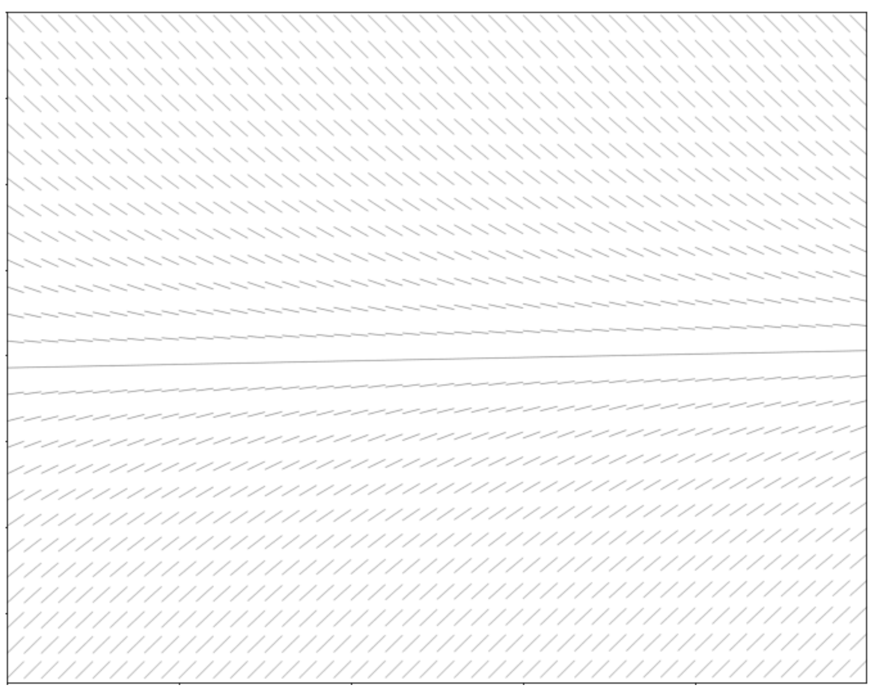
$f(t)$: piecewise constant graph
$v(t)$: piecewise linear graph
$x(t)$: piecewise quadratic graph

Problem 4.6

Part a

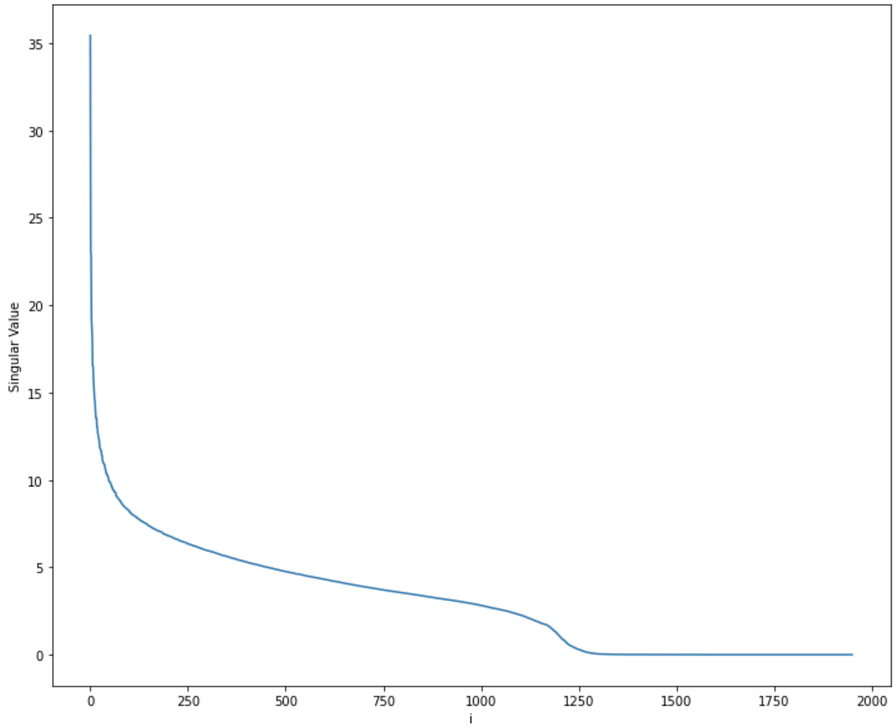


Scan of Image A



Scan of Image -A

Part b



The hidden message is 'WINTER IS COMING'

Python Code

```
import numpy as np
import matplotlib.pyplot as plt

import matlab.engine

matlab_engine = matlab.engine.start_matlab()

height = 50
width = 50
voxels = height*width
m = 1950

A = np.zeros((m,voxels))
x = np.zeros

for i in range(0,voxels,1):
    x = np.zeros(voxels)
    x[i] = 1
    x = x.reshape(height, width)

    scan = matlab_engine.scanImage(matlab.int8(x.tolist())).reshape(m)
    A[:,i] = np.array(scan)

plt.figure()
plt.imshow(A, cmap='gray')

plt.figure()
plt.imshow(-A, cmap='gray')

U, S, V_t = np.linalg.svd(A)
S_inv = 1/np.array(S)

plt.figure()
plt.xlabel('i')
plt.ylabel('Singular Value')
plt.plot(S)

cutoff = np.diag(S_inv[0:1250])
dot = np.dot(V_t[0:1250,:].T, cutoff)
A_inv = np.dot(dot, U[:,0:1250].T)

scan = np.array(matlab_engine.scanImage()).reshape(m)
x = A_inv.dot(scan)
M_un = x.reshape(height, width)
```

```
plt.figure()
plt.imshow(M_un, cmap='gray')
```