PERSONAL NOTES

Linear Algebra

Hanfeng Zhai

Disclaimer: These notes are intended solely for personal reference and study purposes. They represent my own understanding of the course material and may contain errors or inaccuracies. The content presented here should not be considered as an authoritative source, and reliance solely on these materials is not recommended. If you notice any materials that potentially infringe upon the copyright of others, please contact me at hz253@cornell.edu so that appropriate action can be taken. Your feedback is greatly appreciated.

2023

Week 1, 1&2

Date.

No.

Basic Concepts

*

a

$\vec{v}$

$\underset{\sim}{A}$

$$\vec{V} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{bmatrix}$$

Column vec.

↑

m Dimension

m×1

$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots \\ A_{21} & & \\ \vdots & & \end{bmatrix}$$

n×n

$\boxed{\text{collections of vectors}}$

n×n Dimension.

$$a_{\cdot j} = \begin{bmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{nj} \end{bmatrix} = \vec{a}_j \qquad \vec{r}_i = \begin{bmatrix} r_{i1} \\ r_{i2} \\ \vdots \\ r_{in} \end{bmatrix}$$

↑

"it's a row, but looks like a col."

☆ think of all vectors as columns.

Date.

No.

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & & & \vdots \\ \vdots & & \ddots & \\ a_{2m} & & \cdots & a_{mm} \end{bmatrix} \quad (m \times m).$$

$$A^T = \begin{bmatrix} a_{11} & a_{21} & \cdots & a_{m1} \\ a_{12} & a_{22} & & \vdots \\ \vdots & & & \\ a_{1m} & & \cdots & a_{mm} \end{bmatrix}$$

Simple linear system.

$$\begin{cases} 2x_1 + 3x_2 - x_3 = 4. \\ x_1 + 2x_3 = 3 \\ 2x_2 + 3x_3 = 5 \end{cases}$$

$$\begin{cases} 2x_1 + 3x_2 - 1x_3 = 4. \\ 1x_1 + 0x_2 + 2x_3 = 3 \\ 0x_1 + 2x_2 + 3x_3 = 5 \end{cases} \qquad \vec{X} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

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

Date.

No.

$$\vec{b} = \begin{bmatrix} 4 \\ 3 \\ 5 \end{bmatrix} \quad \Rightarrow . \quad \underset{\sim}{A}\,\vec{x} = \vec{b}$$

$$b_i = \vec{r}_i\,\vec{x}$$

$$\hookrightarrow b_1 = 2x_1 + 3x_2 - x_3 = 4$$

$$\boxed{\text{BLDG} \quad 120-59}$$

Date.
No. Week 2 - 1

operations.

▷ Scalars $\overset{\text{op.}}{\longleftrightarrow}$ scalars

▷ Vectors. · $\vec{x}+\vec{y}$ ← same w/ original.

※

· product. $\vec{x}\,\vec{y}$

↓

$\vec{x}^T\vec{y} = [\quad]\begin{bmatrix} \\ \end{bmatrix}$

⇓

dot product.

result is a scalar.

▷ Matrices.

linear equation systems

$\underset{\sim}{A}\,\underline{x} = \underline{b}$

vector mul. leads to scalar

$$\underset{\sim}{A}\,\underline{x} = \begin{bmatrix} \quad & \quad \end{bmatrix} = \begin{bmatrix} \underline{r}_1\,\underline{x} \\ \underline{r}_2\,\underline{x} \\ \underline{r}_3\,\underline{x} \end{bmatrix}$$

Date.
No.

→ organized by rows → i.t.o. rows

→ organized by columns → in terms of unknowns

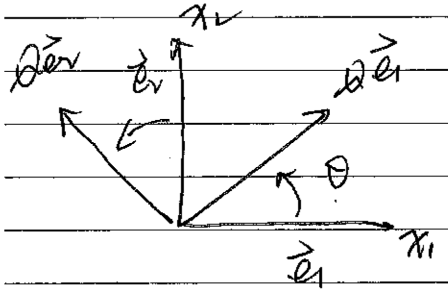
↓

$$\underbrace{\vec{a}_1\underline{x}_1 + \vec{a}_2\underline{x}_2 + \vec{a}_3 x_3}_{\text{columns}}$$

Example. $\underset{\sim}{A}\vec{x} = \vec{b}$

e.g. rotational matrix.

$$\vec{e}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \vec{e}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$



↓

$$Q\vec{e}_1 = \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}; \quad Q\vec{e}_2 = \begin{bmatrix} -\sin\theta \\ \cos\theta \end{bmatrix}$$

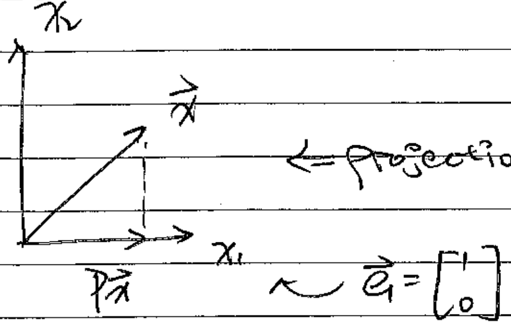
→ matrix multiplication by column.

$$Q\vec{e}_1 = 1\vec{q}_1 + 0\vec{q}_2 = \vec{q}_1 = \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}$$

$$Q\vec{e}_2 = 0\vec{q}_1 + 1\vec{q}_2 = \vec{q}_2 = \begin{bmatrix} -\sin\theta \\ \cos\theta \end{bmatrix}$$

We have hence determined Q:

$$Q=\begin{bmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{bmatrix}$$



$\vec{x}=\begin{bmatrix}x_1\\x_2\end{bmatrix}$.

$$P\vec{x}=\vec{p}_1x_1+\vec{p}_2x_2=\begin{bmatrix}P_{11}\\P_{12}\end{bmatrix}x_1+\begin{bmatrix}P_{21}\\P_{22}\end{bmatrix}x_2$$

$$=\vec{e}_1x_1$$

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

$\vec{x}$ & $P\vec{x}$ different direction & length!

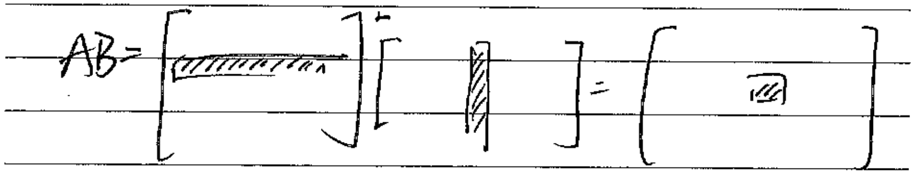
Vector - vector & matrix - vector product.

▷ linear combination

$$c_1\vec{a}_1+c_2\vec{a}_2+\cdots c_n\vec{a}_n=\sum_{j=1}^{n}c_j\vec{a}_j$$

$$=\begin{bmatrix}c_ja_{1j}\\c_ja_{2j}\\\vdots\\c_ja_{nj}\end{bmatrix}$$

▷ Matrix-Matrix Multiplication



$$=\begin{bmatrix}\vec{r}_1^T\vec{b}_1 & \vec{r}_1^T\vec{b}_2 & \cdots & \vec{r}_1^T\vec{b}_5\\ \vec{r}_2\vec{b}_1 & & \cdots & \vdots\\ \vdots & & \ddots & \\ \vec{r}_6^T\vec{b}_1 & & \cdots & \vec{r}_6^T\vec{b}_5\end{bmatrix}$$

Definition

For scalars : $ab=ba$

Date.

No.

"order matters!"

For matrices A, B. (assume square)

$\Downarrow$

$AB \neq BA$ (does not commute in general).

not guaranteed

Pre- & Post- Multiplication.

☆ premultiply : operating on rows AB

post multiply : operating on columns BA

$\underset{\text{post}}{AB}$ $\underset{\text{pre}}{BA}$.

* TAL

- Identity Matrix ; - Permutation Matrix.

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \longleftrightarrow \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$AP \neq PA \neq A$

▷ Matrix Inverse.

$ax = b \rightarrow x = b/a$.

$A\vec{x} = \vec{b} \rightarrow \vec{x} = A^{-1}\vec{b}$

Date.

No.

Definition:

$$AA^{-1} = I \quad \& \quad A^{-1}A = I$$

Find inverse:

using definition: $AA^{-1} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}\begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

$$\begin{cases} a+2c=1 \\ 3a+4c=0 \\ b+2d=1 \\ 3b+4d=0 \end{cases}$$

$A^{-1} = \begin{bmatrix} -2 & 1 \\ \frac{3}{2} & -1/2 \end{bmatrix}$ solve it $\leftarrow$

↓ test it.

$A^{-1}A = I$.

* both statements have to be true.

Rotation Matrix.

$Q = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \rightarrow QQ^{T} = I$

$Q^{-1}Q = I$.

$Q^{-1} = \begin{bmatrix} \cos(-\theta). & -\sin(-\theta) \\ \sin(-\theta) & \cos(-\theta) \end{bmatrix}$

Projection - does not exist. Cannot find

A^{-1} not exist → A non-singular.

Proof: A^{-1} exists IFF $A\vec{x}=\vec{b}$ has a unique solution.

two proofs

↳ both are valid

⇒ A^{-1} exists, → $A^{-1}A=I$.

⇒ $A\vec{x}=\vec{b}$

$A^{-1}(A\vec{x})= A^{-1}\vec{b}$.

$\vec{x}=A^{-1}\vec{b}$.

if $\vec{y}$ is also a solution:

$\vec{y}=A^{-1}\vec{b}$.

$\vec{y}=\vec{x}$ the sol'n is unique.

If $\vec{b}$ has a unique $\vec{x}$. then A^{-1} exist.

Assume $\vec{e}_1=\begin{bmatrix}1\\0\end{bmatrix}$, $\vec{e}_2=\begin{bmatrix}0\\1\end{bmatrix}$

↳ $A^{-1}=[\vec{c}_1\ \vec{c}_2]$.

$AA^{-1}=A[\vec{c}_1\ \vec{c}_2]=\begin{bmatrix}1&0\\0&1\end{bmatrix}$.

$$\Rightarrow \begin{cases}A\vec{c}_1=\vec{e}_1\\ A\vec{c}_2=\vec{e}_2\end{cases}$$

Date.

No.

Week 2 -2.

Definitions { notations, scalar, vec, matrices
col & row
⋮

Norm. $\vec{x} \in \mathbb{R}^m$

compare objects

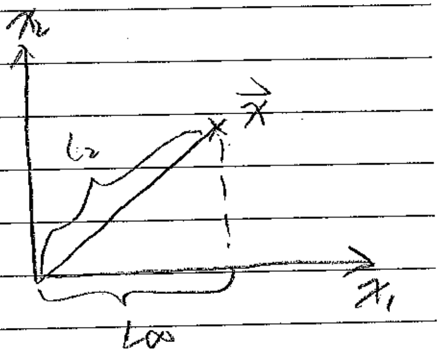
Vector Norms

Euclidean Norm $L_2 = \|\vec{x}\|_2$

Maximum Norm $L_\infty = \|\vec{x}\|_\infty$

Taxi-Cab

Distance between Vectors



$d(x,y) = \|\vec{x} - \vec{y}\|_2$

$= \sqrt{\sum_i (x_i - y_i)^2}$

Matrix Norms

$A\vec{x}' \in \mathbb{R}^{m\times n}$

Frobenius Norm: $L_F = \|A\|_F = \sqrt{\sum_j \sum_i a_{ij}^2}$

Date.

No.

Induced Norms

$\|A\|_2 = \max_{\forall \vec{x} \neq 0} \frac{\|A\vec{x}\|_2}{\|\vec{x}\|_2}$ → effect of matrix on a vector

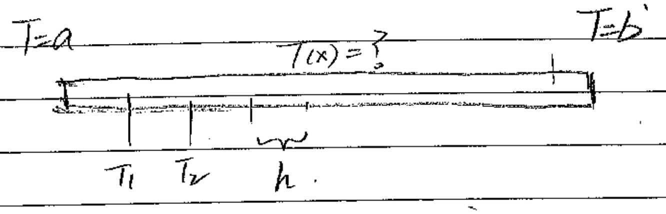
→ not easy to compute, → try all vectors.

$\|A\|_\infty = \max \sum_{j=1}^{n} |a_{ij}|$ → maximum abs row sum of A

$\|A\|_1 = \max \sum_{i=1}^{m} |a_{ij}|$ → ~ column sum of A

Properties of norms $\|\cdot\|$

- $\|A\| \geq 0$, $\|\vec{x}\| \geq 0$
- $\|A\| = 0$ IFF $A = 0$ & $\|\vec{x}\| = 0$ IFF $\vec{x} = 0$
- $\|\alpha A\| = |\alpha| \|A\|$ and $\|\alpha \vec{x}\| = |\alpha| \|\vec{x}\|$
- $\|A + B\| \leq \|A\| + \|B\|$ & $\|\vec{x} + \vec{y}\| \leq \|\vec{x}\| + \|\vec{y}\|$
- $\|AB\| \leq \|A\| \|B\|$ ~ $\|\vec{x}^T y\| \leq \|\vec{x}\| \|y\|$
- $\|A\vec{x}\| \leq \|A\| \|\vec{x}\|$.



Temperature distribution

$$\vec{T} = \begin{bmatrix} T_1 \\ T_2 \\ \vdots \\ T_n \end{bmatrix}$$

~~build linear system:~~ $A\vec{T} = \vec{c}$

☆ permutation for 4x4 matrix

what's the operation?

which rows & columns to flip

Computational complexity

⇓

theoretical analysis of costs

vector - vector product

$$\vec{x}^T\vec{y} = \sum_i^n x_i y_i \longrightarrow (2n-1) \text{ flop}$$

Solving linear systems

⇓

Gaussian limitations

Words to be encripted $\begin{bmatrix} 3 \\ 13 \\ 5 \end{bmatrix}$

applying shift $\vec{S} = \sigma \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$

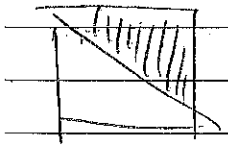
$\sigma = 3$ → secret code

$$\vec{V} + 3\vec{S} = \begin{bmatrix} 3 \\ 13 \\ 5 \end{bmatrix} + 3 \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \vec{w} = \begin{pmatrix} 5 \\ 17 \end{pmatrix}$$

easy to decode.

Gaussian Elimination)

$$A\vec{x} = \vec{b}_1 \Rightarrow U$$



$$A\vec{x} = \vec{b} \xrightarrow{\text{transform}} U\vec{x} = \vec{b}$$

Date.

No.

Gauss Elimination

$$\begin{cases} x_1 - 3x_2 + x_3 = 4 \\ 2x_1 - 8x_2 - 8x_3 = -2 \\ -6x_1 + 3x_2 - 15x_3 = 9 \end{cases} \Rightarrow A\vec{x} = \vec{b}$$

$$A \rightarrow A' \rightarrow A''$$

$$A\vec{x} = \vec{b}$$

$$U\vec{x} = \vec{d} \longrightarrow \vec{x}$$

↳ upper-triangular matrix

* Interprete Gaussian Elimination in a more general sense.

Transformation: $A \rightarrow A' \rightarrow A'' \rightarrow \cdots \rightarrow A^{n-1} = U$

$A \rightarrow A'$, $A' = C_1 A$

- pre-multiplication.

$$A = \begin{bmatrix} & & \\ & & \end{bmatrix} \rightarrow A' = \begin{bmatrix} & & \\ 0 & & \\ 0 & & \end{bmatrix} = C_1 A.$$

an operator

$$C_1 = \begin{bmatrix} 1 & 0 & 0 \\ -a_{21}/a_{11} & 1 & 0 \\ -a_{31}/a_{11} & 0 & 1 \end{bmatrix}$$ C_1 is an operator $= \begin{bmatrix} & & \\ & & \end{bmatrix}$

Date.

No.

$A' \rightarrow A'' \Rightarrow$ interpret this step

$$A' = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a'_{22} & a'_{23} \\ 0 & a'_{32} & a'_{33} \end{bmatrix} \rightarrow A'' = \begin{bmatrix} a'_{11} & a'_{12} & a'_{13} \\ 0 & a'_{22} & a'_{23} \\ 0 & 0 & a'_{33} \end{bmatrix}.$$

$= C_2 A'$

$$C_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -a'_{32}/a'_{22} & 0 \end{bmatrix}.$$

$$U = A'' = C_2(C_1 A)$$

↓

Upper triangular matrix.

- assume C_2 invertible,

$$C_2^{-1} = C_2^{-1} C_2 (C_1 A)$$

$\rightarrow$ ~

$$C_1^{-1} C_2^{-1} U = A$$

The result: $(C_1^{-1} C_2^{-1}) U = LU = A$.

what are C_1^{-1} & C_2^{-1}?

- what is $L = (C_1^{-1} C_2^{-1})$?

check C_1, C_2 inverses

★ Gaussian Elimination in operator form is a factorization of A: $A = LU$

$$A\vec{x} = LU\vec{x} = \vec{b}$$

Solution of linear system $A\vec{x} = \vec{b}$

→ factorization $A = LU$

→ solve $L\vec{y} = \vec{b}$ → get $\vec{y}$

→ solve $U\vec{x} = \vec{y}$. → get $\vec{x}$

LU factorization is essentially Gaussian Elimination

why is LU useful?

because we do not need to change $\vec{b}$

(different from Gaussian Elim.)

Compute inverse $AA^{-1} = I$

$$A \begin{bmatrix} | & | & | \\ | & | & | \\ | & | & | \end{bmatrix} = \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix}$$

↳ decoupled to many linear systems

⇓

LU factorization

prove

$$U = DL^T = \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix} = \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix}\begin{bmatrix} & & \\ & & \\ & & \end{bmatrix}$$

- ∘ $A = LU = LDL^T$
- ∘ $D = \sqrt{D}\sqrt{D}$
- ∘ $A = L\sqrt{D}\sqrt{D}L^T = (L\sqrt{D})(\sqrt{D}L)^T$

$$= \hat{L}\hat{L}^T$$

essentially: applying LU decomposition to both A and A^T

- vecvec
- matvec
- matmat

Date.
No.

backsubstitution. $U\vec{x}=\vec{b}$ compressed

$$x_i = \frac{1}{u_{ii}}\left(b_i - \sum_{j=i+1}^{n} u_{ij}x_j\right)$$

takeaway: the computational complexity for LU factorization is smaller the GE.

$O(n^2)$ $O(n^3)$

matmat → $O(n^3)$

Week 2-3

Gauss Elimination

$$A \sim \begin{bmatrix} 2 & 1 & -1 & 8 \\ 0 & 1 & -0.5 & 2.5 \\ 0 & 0 & 3.5 & 5.5 \end{bmatrix}$$

Unique of the LU Decomposition

for $A=LU$

$$A = L_1U_1 = L_2U_2$$

on the ⇓

row operations

$$L_2^{-1}L_1U_1 = U_2 . \quad L_2^{-1}L_1 = U_2U_1^{-1}$$

Date.
No.

$$L_2^{-1}L_1 = I$$

✓ are the same

by the uniqueness of inverse

1D diffusion equation.

$$a\frac{\partial^2 u}{\partial x^2} = 0$$

⇓ discretize

$$\frac{\partial^2 u_i}{\partial x^2} \approx \frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2}$$

→ mesh size

$$A = \begin{bmatrix} -2 & 1 & & \\ 1 & -2 & 1 & \\ & 1 & -2 & \cdots \\ & & & \ddots \end{bmatrix}$$

In MATLAB

when we have pivots = 0

$A = \sim$

→ compute LU decomposition

$[L, U, P] = lu(A)$

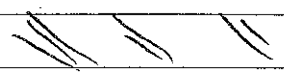
→ permutation matrix

qr(A)

Date.

No.

Spare matrix of A, L, U



Sparsity & LU Decomposition

- Spare Storage Format

Float point representation

Date.

No.

TA session

Q: linear Algebra?

↓

linear combinations.

Define Given $v_1, \ldots, v_m$,

(n-vectors or of $\mathbb{R}^n$)

and $\alpha_1, \ldots, \alpha_m$.

a linear combination of $v_1, \ldots, v_m$ is

a vector of the form

$$w = \alpha_1 v_1 + \alpha_2 v_2 + \cdots + \alpha_m v_m$$

Example.

$$v_1 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, \quad v_2 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$$

$$w = 4v_1 - 2v_2 = \begin{bmatrix} 4 \\ 0 \\ 4 \end{bmatrix} - \begin{bmatrix} 2 \\ 2 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 2 \\ -2 \\ 4 \end{bmatrix}.$$

Big part of what we'll do

explore sets of linear combs.

For example, we focus on solving linear systems. But this is just answering whether a vector, $\vec{b}$, can be expressed as a linear combination of combs. of A.

Mat-vec prod. intepretation

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} \vec{r}_1^T \\ \vec{r}_2^T \\ \vdots \\ \vec{r}_n^T \end{bmatrix}$$

$$= \begin{bmatrix} \vec{a}_1 & \vec{a}_2 & \cdots & \vec{a}_n \end{bmatrix}$$

$$Ax = \begin{bmatrix} \vec{a}_1 & \vec{a}_2 & \cdots & \vec{a}_n \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

$$= x_1\vec{a}_1 + x_2\vec{a}_2 + \cdots + x_n\vec{a}_n$$

Therefore, solving for $Ax = b$ is equivalent to finding coeffs. $x_1, x_2, \ldots, x_n$ s.t.

$$x_1\vec{a}_1 + x_2\vec{a}_2 + \ldots + x_n\vec{a}_n = \vec{b}$$

i.e., s.t. $\vec{b}$ is a lin. comb. of col'ns of A!

① as a motivating example

Every complex # $z = x + iy$.

with $x,y \in \mathbb{R}$, and $i=\sqrt{-1}$,

1). Addition

$$Z_1 + Z_2 = Z_2 + Z_1$$

(comunatativity)

$$(x_1+iy_1)+(x_2+iy_2) = (x_1+x_2)+i(y_1+y_2)$$

$$(Z_1+Z_2)+Z_3 = Z_1+(Z_2+Z_3)$$

(associativity)

$Z_1+0=0+Z_1=Z_1$ (additive identity)

For every complex Z_1, we have

$-Z_1$ and $Z_1+(-Z_1)=0$ (additive inv.)

multiplication

- Complex mul. real number

2) scalar multiplication

Given any $\alpha \in \mathbb{R}$, we can multiply

$$\alpha z_1 = \alpha(x+iy)$$

$$= \alpha x + i(\alpha y)$$

$$\alpha(Z_1+Z_2) = \alpha Z_1 + \alpha Z_2$$

(distribution prop. I)

$$(\alpha_1+\alpha_2)Z_1 = \alpha_1 Z_1 + \alpha_2 Z_1$$

(distribution prop II)

defines a vector space

!!

addition scalar, add: operation.

key takeaway Abstract

nice and well-known properties so we can recognize common structure in a

Date.
No.

myriad of different scenarios

Key Addition and scalar mult. are exactly what's needed to for linear comb.!

Def'n (informal) a vector space is any sets of objects s.t the linear comb. of any finite sets $V_1, V_2, \ldots, V_n$ is an element of V.

Def'n A set V is a real vector space (V.S.) if there is a condition operation + and a scalar mult * s.t.

1) $v + w = w + v$ (commutivity)

2) $(V+W)+u = V+(W+u)$ (associativity)

3) there is an elt. labeled 0, s.t

Date.
No.

$v+0 = 0+v = v$ (additive i.d.)

4) there is v. s.t

$v+(-v) = (-v)+v$ (additive inv.)

5) for any $\alpha, \beta \in \mathbb{R}$, we have

$(\alpha+\beta) v = \alpha \cdot v + \beta \cdot v$

$\alpha * (v + w) = \alpha \cdot v + \alpha \cdot w$

Examples

1) $\mathbb{R}$, $\mathbb{C}$ and $\mathbb{R}^n$

2) $\{0\}$.

Q. How many elmt. can a vs have?

One ($\{0\}$) or infinitely many

3) The set of all $m \times n$ matrices

Zero matrix $\begin{bmatrix} 0 & \cdots & 0 \\ \vdots & & \vdots \\ 0 & \cdots & 0 \end{bmatrix} = \text{zeros}(m,n)$

4) The set of all polynomials of degree up to n.

$$V = \{ \alpha_0 + \alpha_1 x + \cdots + \alpha_n x^n \mid \alpha_j \in \mathbb{R} \}$$

Take

$$P_1 = \alpha_0 + \alpha_1 x + \cdots + \alpha_n x^n$$

$$P_2 = \beta_0 + \beta_1 x + \cdots + \beta_n x^n$$

$$P_1 + P_n = (\alpha_0 + \beta_0) + (\alpha_1 + \beta_1) x + \cdots + (\alpha_n + \beta_n) x^n$$

Zero elemt

$$0 = \underset{\mathbb{R}}{\underset{\downarrow}{0}} = 0 + 0x + 0x^2 + \cdots + 0x^n$$

Scalar multi

$$4P_1 = 4\alpha_0 + (4\alpha_1) x + \cdots + (4\alpha_n) x^n$$

V.S. related matrices.

Given an $m \times n$ matrix A

1) Define the col'n space of A

col(A) is the set of all lin. comb. of col'ns of A.

e.g. $A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}$

$$\text{col}(A) = \left\{ \alpha \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} + \beta \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \middle| \alpha, \beta \in \mathbb{R} \right\}.$$

$$= \left\{ \begin{bmatrix} \alpha + \beta \\ \beta \\ \alpha \end{bmatrix} \middle| \alpha, \beta \in \mathbb{R} \right\}.$$

2) Def'n The row space of A, row(A) is set of all lin. comb. of rows of A.

e.g. A as above

$$\text{row}(A) = \left\{ \alpha \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \beta \begin{bmatrix} 0 \\ 1 \end{bmatrix} + \gamma \begin{bmatrix} 1 \\ 0 \end{bmatrix} \middle| \alpha, \beta, \gamma \right\}$$

Date.

No.

$$= \left\{ \begin{bmatrix} \alpha+\gamma \\ \alpha+\beta \end{bmatrix} \middle| \ \alpha, \beta, \gamma \in \mathbb{R} \right\}$$

3) Defn the nullspace of A, $\text{null}(A)$

is

$$V = \{ x \mid Ax = 0 \}$$

Exercise: why this indeed is a v.s.

Q

How to show our set is a v.s.?

A: If you recognize your set is a subset of a known v.s. then all you need is

1). zero elt, belongs to your set!

2) for any x, y, in your set, $\alpha x + \beta y$

?...

Date.

No.

Week 4 - 1

$$Az = A(\alpha x + \beta y)$$

$$= \alpha(Ax) + \beta(Ay)$$

$$= \alpha \cdot 0 + \beta \cdot 0$$

$$= 0$$

Example

Given $m \times n$ A, check that

$$\mathcal{N}(A) = \{ x \in \mathbb{R}^n \mid Ax = 0 \}$$

First notice $\mathcal{N}(A) \subseteq \mathbb{R}^n$; every x s.t. $Ax = 0$ an n-vector in $\mathbb{R}^n$! Now notice that

$$A0 = \begin{bmatrix} & \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Now we've shown that any linear combinations of x, y. also belongs to the null space $\mathcal{N}(A)$. So we also have

$\mathcal{N}(A)$ is a vs.

Consequences

homogeneous linear system.

Since de nullspace is a vector space.
N(A)

→ non-empty. → $Ax=0$ always has at least one solution, namely the zero vector. $\vec{0}$.

(or closure prop.)

→ the set is closed under linear combination, i.e., if you find a non-zero sol'n, there has to be infinitely many (distinct sol'n)

then, there should be infinitly many zero vec.!

Example:

Say, $A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}$

Notice that $A\begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}\begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

So, $\begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix} \in N(A)$ and therefore

$\begin{bmatrix} 2 \\ -2 \\ 0 \end{bmatrix}, \begin{bmatrix} \pi \\ -\pi \\ 0 \end{bmatrix}, \begin{bmatrix} 1/3 \\ -1/3 \\ 0 \end{bmatrix}$, etc., also belong to $N(A)$.

Definition Given a list, $v_1, \dots, v_m$ of vectors in V, the span of $v_1, \dots, v_m$ is defined as

$$\text{Span}(v_1, \dots, v_m) = \{\alpha_1 v_1 + \dots + \alpha_m v_m \mid \alpha_1, \dots, \alpha_m \in \mathbb{R}\}$$

(set of all lin. comb. of $v_1, \dots, v_m$)

Example

From last time, given an mxn A,

$Col(A) = \{\alpha_1 \vec{a}_1 + \cdots \alpha_n \vec{a}_n \mid \alpha_1, \ldots, \alpha_n \in \mathbb{R}\}$

(col. space of A set of lin. comb of the colms of A)

Now,

$$col(A) = span(\vec{a}_1, \vec{a}_2, \cdots, \vec{a}_n).$$

Also,

$$row(A) = span(\vec{r}_1, \vec{r}_2, \cdots, \vec{r}_m).$$

$$A \begin{bmatrix} - \vec{r}_1 - \\ \vdots \\ - \vec{r}_m - \end{bmatrix}$$

Fact.

For any $v_1, \ldots, v_m$ in a v.s. V,

$span(v_1, \ldots, v_m)$ is always a v.s..

P.f. (Sketch).

1) Contains all possible lin. comb, so

in particular we can set $\alpha_1 = \cdots = \alpha_m = 0$,

and note:

$$0 v_1 + \cdots + 0 v_m = 0 \in span(v_1, \ldots, v_m)$$

2) lin. comb. are included by definition.

so in particular $\alpha x + \beta y \in span(v_1, \ldots, v_m)$

for any $x, y \in span(v_1, \ldots, v_m)$

Fact: In fact, $span(v_1, \ldots, v_m)$ is the smallest v.s. containing $v_1, \ldots, v_m$.

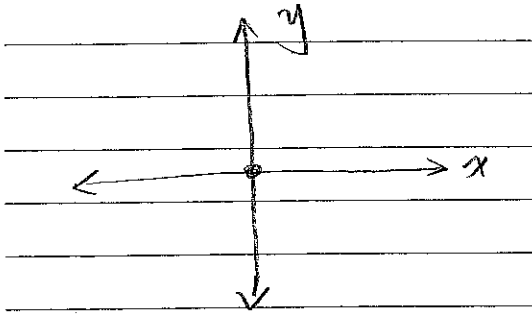
Key: For any (given) vector space, V, I can always find a list $v_1, \ldots, v_m$, s.t.,

$$V = span(v_1, \ldots, v_m)$$

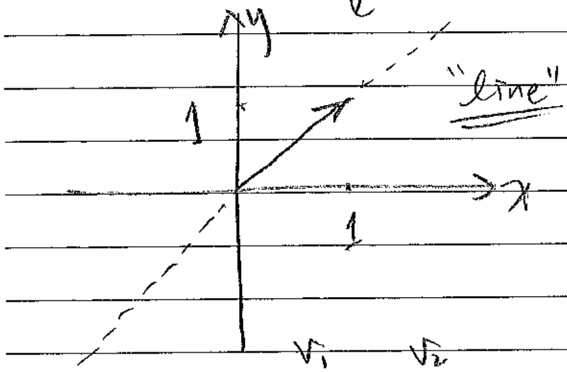
(spanning list / set)

Example (Geometric interpretation of span)

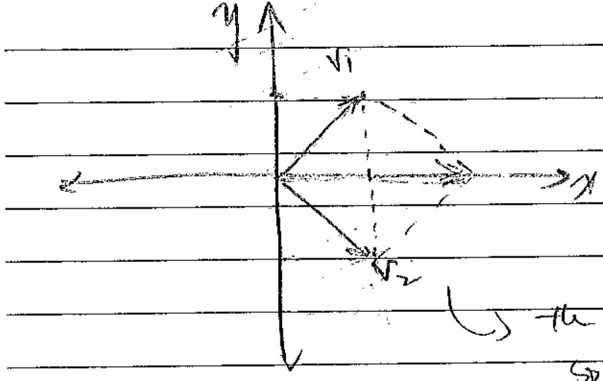
$$span\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}\right) = \left\{\begin{bmatrix} 0 \\ 0 \end{bmatrix}\right\}$$



$\Rightarrow \text{span}\left(\begin{bmatrix}1\\1\end{bmatrix}\right) = \left\{\begin{bmatrix}\alpha\\\alpha\end{bmatrix} \middle| \alpha \in \mathbb{R}\right\}$



$\rightarrow \text{span}\left(\begin{bmatrix}1\\1\end{bmatrix}, \begin{bmatrix}1\\-1\end{bmatrix}\right) \rightarrow \begin{bmatrix}\alpha+\beta\\\alpha-\beta\end{bmatrix}$



↳ the whole plane

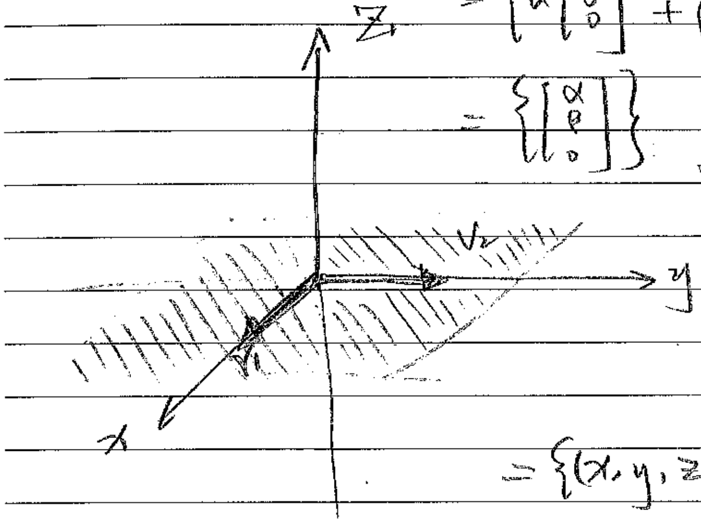
span $\in \mathbb{R}^2$

example

$$\text{span}\left(\begin{bmatrix}1\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\0\end{bmatrix}\right) = xy\text{- plane}$$

$$= \left\{\alpha\begin{bmatrix}1\\0\\0\end{bmatrix} + \beta\begin{bmatrix}0\\1\\0\end{bmatrix} \middle| \alpha,\beta \in \mathbb{R}\right\}$$

$$= \left\{\begin{bmatrix}\alpha\\\beta\\0\end{bmatrix}\right\}$$



$$= \{(x,y,z) \mid z=0\}$$

Linear Independence.

Saying that every v.s. has a spanning list means that for any $\underline{v}$ in a given v.s. V. (so say $V = \text{span}(v_1, \ldots, v_m)$) I can always find scalars $c_1, \ldots, c_m$, s.t.,

$$V = c_1v_1 + \ldots + c_mv_m$$

Q: Is there only one way to derive v? A: not always.

Suppose you find other scalars, $d_1, \dots, d_m$ s.t.

$$v = d_1 v_1 + \cdots + d_m v_m.$$

So, we have

$$c_1 v_1 + \cdots + c_m v_m = v$$

$$= d_1 v_1 + \cdots + d_m v_m.$$

This is equivalent to saying that

$$(c_1 - d_1) v_1 + \cdots + (c_m - d_m) v_m = 0$$

$$\Downarrow$$

* Implied linear independence ...?

Note: if $c_1 = d_1 = \cdots = c_m = d_m = 0$, then actually description is unique!

Defn A list $v_1, \dots, v_m$ is

linearly independent (lin. ind) if the only linear combination of $v_1, \dots, v_m$:

$$\alpha_1 v_1 + \cdots + \alpha_m v_m$$

resulting in zero, is $\alpha_1 = \cdots = \alpha_m = 0$

Reminder

HW2 - 10/20

Exam. - 10/25

Monday.

- Subspaces, span, lin. ind.

Today.

- Basis, dimension, rank.
- Determinants.

Defn

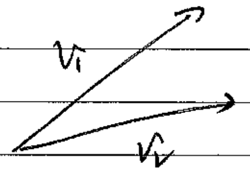
A list of vecs $v_1, \dots, v_m$ in a v.s. V is lin. ind. if the only linear combination of $v_1, \dots, v_m$ resulting in $\vec{0}$ is the trivial comb. (every coeff. is zero. In other words, $v_1, \dots, v_m$ is linearly independent).

$$\alpha_1 v_1 + \dots + \alpha_m v_m = 0.$$

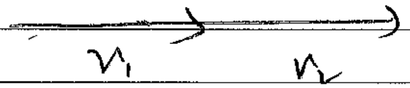
implies $\alpha_1 = \dots = \alpha_m = 0$.

Example.

- If $v_1 \neq 0$, then v_1 is lin. ind.
- Two vecs v_1, v_2 are lin. ind., if they are **not** scalar mult. of each other.



are lin. ind.



continued

Week 3 - 1

Polynomial Interpolation

$$y = \alpha_1 + \alpha_2 x + \dots + \alpha_n x^{n-1}$$

$$= \sum_{j=1}^{n} \alpha_j x^{j-1}$$

Problem: find α_j

from 4 observations cubic

$$\begin{cases} y_1 = \alpha_1 + \alpha_2 x_1 + \alpha_3 x_1^2 + \alpha_4 x_1^3 \\ y_2 = \quad \dots \\ y_3 = \\ y_4 = \end{cases}$$

$$\Downarrow$$

build cubic interpolant: $y = \alpha_1 + \alpha_2 x + \alpha_3 x^2 + \alpha_4 x^3$

problem: $\begin{bmatrix} 1 & 1 & 1^2 & 1^3 \\ 1 & 8 & 8^2 & \\ 1 & 10 & 10^2 & \\ 1 & 16 & 16^2 & \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 8 \\ 8 \\ 16 \end{bmatrix} \rightarrow A\vec{\alpha} = \vec{y}$

use Gauss Elimination.

$$\vec{\alpha} = \begin{bmatrix} \\ \\ \end{bmatrix}$$

$$A = \begin{bmatrix} & & \\ & & \\ \end{bmatrix} \rightarrow \dots \rightarrow \begin{bmatrix} \ddots & & \\ 0 & \ddots & \\ 0 & 0 & \ddots \end{bmatrix}$$

consider system:

$$A\vec{x} = \begin{bmatrix} & \\ & \end{bmatrix} \vec{x} = \vec{b} = \begin{bmatrix} \\ \end{bmatrix}$$

when Gaussian Elimination does not work

$$\Downarrow$$

Switch the rows

* We can use permutation matrix to swap rows

for $A = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 5 \\ 4 & 6 & 8 \end{bmatrix}$ $P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$

Product $PA = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 5 \\ 4 & 6 & 8 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 4 & 6 & 8 \\ 2 & 2 & 5 \end{bmatrix}$

$$PA = \begin{bmatrix} 1 & 1 & 1 \\ 4 & 6 & 8 \\ 2 & 2 & 5 \end{bmatrix} \rightarrow A' = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 2 & 4 \\ 0 & 0 & 3 \end{bmatrix}$$

Date.
No.

Pivoting is not unique.

$$\hat{P} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \circlearrowright \text{ swap 1st \& 3rd row}$$

$$\hat{P}A = \begin{bmatrix} 4 & 6 & 8 \\ 2 & 2 & 5 \\ 1 & 1 & 1 \end{bmatrix}$$

↑

swap ~~the~~ 1st & 3rd row

if no permutation enables Gauss elimination, the matrix A is not invertible

Given system of equations

$$A\vec{x} = \vec{b}, \quad A \in \mathbb{R}^{n\times n}$$

GE transforms: $A \to A' \to A'' \to \cdots \to U$

"With pivoting" requires additional step:

$$A \to P_1A \to A' \to P_2A' \to A'' \to \cdots \to U.$$

1st step: $P_1A \to A' \equiv A' = C_1P_1A$

Date.
No.

2nd step: $P_2A' \to A'' \equiv A'' = C_2P_2A'$

$$= C_2P_2(C_1P_1A)$$

How do we proceed to $A = LU$?

Assume $C_1^{-1}, C_2^{-1}, \dots$

$$A' = C_1A \to C_1^{-1}A' = A$$

$$A'' = C_2A' \to C_2^{-1}A'' = A'$$

equivalent

$$C_1^{-1}C_2^{-1}A'' = C_1^{-1}A' = A$$

Assume $(C_1P_1)^{-1}, (C_2P_2)^{-1}, \dots$

$$A' = C_1P_1A \to (C_1P_1)^{-1}A' = A$$

$$A'' = C_2P_2A' \to (C_2P_2)^{-1}A'' = A'$$

What is the inverse of a permutation mat.

What is inverse of $(C_1P_1)^{-1}$ < $P^{-1} = P$

↳ $(C_1P_1)^{-1} = P_1^{-1}C_1^{-1} = P_1C_1^{-1}$ "row swap"

Operator perspective

$$A' = C_1 P_1 A \rightarrow (C_1 P_1)^{-1} A'$$

$$= P_1 C_1^{-1} A' = A.$$

Premultiply P_1^{-1} : $P_1^{-1} P_1 C_1^{-1} A' = C_1^{-1} A'$

$$= P_1^{-1} A = P_1 A$$

First step of GE pivoting

① $C_1^{-1} A' = P_1 A$

↳ ② $C_2^{-1} A'' = P_2 A'$

if you follow all the steps

$$(C_1^{-1} C_2^{-1} \cdots C_{n-1}^{-1}) U = (P_n P_{n-1} \cdots P_1) A$$

$$LU = PA \qquad \#$$

↓

row swaps

(permutation)

Zero pivots can be resolved w/ pivoting

$$\hat{P}A$$

Another issue – limited precision

System $A = \begin{bmatrix} 0.01 & 1 \\ 1 & -1 \end{bmatrix}$ $b = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

exact solution $= \begin{bmatrix} 100/101 \\ 100/101 \end{bmatrix}$

numbers are stored under different digits precision.

↳ { single precision
double precision
half precision.

▷ GE w/ high precision

$$U = \begin{bmatrix} 0.01 & 1 \\ 0 & -101 \end{bmatrix} \quad \vec{d} = \begin{bmatrix} 1 \\ -100 \end{bmatrix} \rightarrow \vec{x} \triangleq \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

▽ GE w/ low precision

$$U_{sd} = \begin{bmatrix} 1.0E{-2} & 1.0E0 \\ 0.0E0 & -1.0E2 \end{bmatrix} \quad \ldots \quad \vec{x}_{sd} = \begin{bmatrix} 0.0E0 \\ 1.0E0 \end{bmatrix}$$

pivoting with low precision

$$U_{psd} = \begin{bmatrix} \quad \end{bmatrix} ; \quad \vec{d}_{psd} = \begin{bmatrix} \quad \end{bmatrix}$$

↓

$$\vec{x}_{psd} = \begin{bmatrix} 1.0E0 \\ 1.0E0 \end{bmatrix} ✓$$

pivoting can reduce the propagation & growth of the numeartion error.

Ill-Conditioned System

$$\begin{cases} x_1 + 2x_2 = 3 \\ 3x_1 - 2x_2 = 1 \end{cases} \qquad \begin{cases} x_1 + 2x_2 = 3 \\ 3x_1 - 2x_2 = 1.008 \end{cases}$$

$$\vec{x} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \qquad \vec{x} = \begin{bmatrix} \quad \end{bmatrix} \approx \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

the system is very sensitive to permbations.

Ill-conditioned. small noises induce / error lary response changes

Consider: $A\vec{x} = \vec{b}$

$$A\vec{y} = \vec{b} + \delta\vec{b}$$

In other words: $\vec{y} \approx \vec{x}$, $y = \vec{x} + \delta\vec{x}$

$$\|\delta\vec{b}\| / \|\vec{b}\| \Rightarrow \|\delta\vec{x}\| / \|\vec{x}\|$$

Small

Not very useful to check solutions!

$$A\vec{x} = \vec{b} \leftarrow \text{original}$$

$$A(\vec{x} + \delta\vec{x}) = (\vec{b} + \delta\vec{b}) \leftarrow \text{perturbed}$$

Substracting $A\delta\vec{x} = \delta\vec{b}$

$$\Downarrow$$

$$\delta\vec{x} = A^{-1}\delta\vec{b}$$

"norms":

$$\|\delta\vec{x}\| = \|A^{-1}\delta\vec{b}\| \leq \|A^{-1}\|\|\delta\vec{b}\|$$

$$\|\vec{b}\| = \|A\vec{x}\| \leq \|A\| \|\vec{x}\|$$

equalent to:

$$\frac{1}{\|\vec{x}\|} \leq \frac{\|A\|}{\|\vec{b}\|}$$

compute:

$$\frac{\|\delta\vec{x}\|}{\|\vec{x}\|} \leq \|A^{-1}\|\|\delta\vec{b}\| \frac{1}{\|\vec{x}\|}$$

Date.
No.

$\rightarrow \; ||| \; \rightarrow$

$\downarrow$

$$\frac{\|\varepsilon \vec{x}\|}{\vec{x}} \leq (\|A^{-1}\| \|A\|)$$

✓

Conditional number

~~large condition → positively correlated with sensitivity~~

~~high condition number~~

↓

more sensitive to the environment

large Co. num. → matrix close to singular

→ why?

— why if num small?

MATLAB → condition number using function

Date.
No.

~~example. change from (10,8) → (9,9)~~

Week 4 -2.

$$- \begin{bmatrix}1\\0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\\0\end{bmatrix} \text{ is lin. ind. in } \mathbb{R}^4.$$

Q: Why? How to show lin. ind.?

A: say there are coeff.

s.t.

$$c_1\begin{bmatrix}1\\0\\0\\0\end{bmatrix} + c_2\begin{bmatrix}0\\1\\0\\0\end{bmatrix} + c_3\begin{bmatrix}0\\0\\1\\0\end{bmatrix} = \begin{bmatrix}0\\0\\0\\0\end{bmatrix}.$$

$$\hookrightarrow \begin{bmatrix}c_1\\c_2\\c_3\\0\end{bmatrix} = \begin{bmatrix}0\\0\\0\\0\end{bmatrix}$$

The last eq. implies $c_1 = c_2 = c_3 = 0$.

So vecs. are lin. ind. by def'n

$$\begin{array}{ccc} (4\times 3) & (3\times 1) & (4\times 1) \\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = & \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \end{array}$$

$$\Downarrow$$

$$\begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ 0 \end{bmatrix}$$

Note: Deciding whether a given list is lin. ind. is equivalent to counting # of sol'ns to a homogenous system. ~~to~~ determining null space of an associated matrix.

Defn The vecs $v_1, \dots, v_m$ are

☆ Cols are linearly independent. null space just the zero vector (singular...)

linearly dependent if they are not lin. ind.

Roughly, this means one vec is a lin. comb. of the others.

Example.

$$\begin{bmatrix} 2 \\ 3 \\ -1 \end{bmatrix}, \begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix}, \begin{bmatrix} 7 \\ 3 \\ 8 \end{bmatrix}$$
$$\quad\ \ \| \qquad\quad \| \qquad\quad \|$$
$$\quad\ v_1 \qquad\ v_2 \qquad\ v_3.$$

Note: $2v_1 + 3v_2 - v_3 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = \vec{0}$

there's a way ↗ to get the zero vec.

$$\Downarrow$$

Not linearly independent.

i.e. linearly dependent

Date.
No.

Also note Rearrange last eq. to obtain

$$V_3 = 2V_1 + 3V_2.$$

Q: How do I find these coeffs. e.g. (2, -3, 1)?

A: Solve a linear system!

I want to find c_1, c_2, c_3, s.t.

$$c_1 v_1 + c_2 v_2 + c_3 v_3 = \vec{0}$$

$$c_1 \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix} + c_2 \begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix} + c_3 \begin{bmatrix} 7 \\ 3 \\ 8 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 2c_1 + c_2 + 7c_3 \\ 3c_1 + (-1)c_2 + 3c_3 \\ 1c_1 + 2c_2 + 8c_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$\vec{c}$

Notice, e.g.,

$$2c_1 + 1 \cdot c_2 + 7c_3 = \overbrace{\begin{bmatrix} 2 & 1 & 7 \end{bmatrix}}^{r_1^T} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

Date.
No.

Similarly,

$$3c_1 + (-1)c_2 + 3c_3 = \overbrace{\begin{bmatrix} 3 & -1 & 3 \end{bmatrix}}^{\vec{r}_2^T} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}.$$

Therefore,

$$c_1 \underbrace{\begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix}} + c_2 \underbrace{\begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix}} + c_3 \underbrace{\begin{bmatrix} 7 \\ 3 \\ 8 \end{bmatrix}} = \begin{bmatrix} \vec{r}_1^T \vec{c} \\ \vec{r}_2^T \vec{c} \\ \vec{r}_3^T \vec{c} \end{bmatrix}$$

$$= \begin{bmatrix} - r_1^T - \\ - r_2^T - \\ - r_3^T - \end{bmatrix} \vec{c}$$

$$= \begin{bmatrix} 2 & 1 & 7 \\ 3 & -1 & 3 \\ 1 & 2 & 8 \end{bmatrix} \vec{c}$$

Key Point. Set $A = \begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix}$

Then,

$$c_1 \vec{v}_1 + c_2 \vec{v}_2 + c_3 \vec{v}_3 = A \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = A\vec{c}$$

Mat-vec prod: is just linear comb of mat col'ns

Back to independence

Given v_1, v_2, v_3, set up.

$$A = [v_1, v_2, v_3] = \begin{bmatrix} 2 & 1 & 7 \\ 3 & -1 & 3 \\ 1 & 2 & 8 \end{bmatrix}$$

and solve $A\vec{c} = 0$ to determine whether v_1, v_2, v_3 are in fact lin. ind. (solve using Gauss Elim. LU, ...)

v. lin. depedent. → A is singular.

Conclusion: Two cases.

1) only soln is $\vec{c} = 0$. In this case, colmns are lin. ind. and if A is square then it is non-singular (or invertible). ✱ Concepts IMPORTANT

Singularity: only talk about in square matrix

2) Infinitely many sol'n colmns are dependent.

$\vec{C} A$

↳ linear independence of rows

Basis

Defn: A list $v_1, \dots, v_n$ of vecs in V is a basis of V if

1). $v_1, \dots, v_n$ spans V. (span$(v_1, \dots v_n)$)

2) $v_1, \dots, v_n$ is lin. ind.

Interpretation

1) $v_1, \dots, v_n$ reaches all points in V, i.e., every element of V is

a lin. comb. of $v_1, \dots, v_m$

Alternatively, If $v_1, \dots, v_m \in \mathbb{R}^n$, then spanning $\mathbb{R}^n$ means the linear sys.

$$A\vec{x} = \vec{b}$$

has a sol'n for every $\vec{b}$, with $A = [v_1, \dots, v_m]$ has a sol'n for every $\vec{b}$.

2) description in terms of lin. comb. of $v_1, \dots, v_m$ is minimal, i.e., there is no redundancy in list.

If $v_1, \dots, v_m$ are elts. of $\mathbb{R}^n$, then the elmts.

$$A\vec{x} = 0.$$

has a <u>unique</u> sol'n.

Example:

• $\overset{e_1}{\begin{bmatrix}1\\0\\0\\\vdots\\0\end{bmatrix}}, \overset{e_2}{\begin{bmatrix}0\\1\\0\\\vdots\\0\end{bmatrix}}, \dots, \overset{e_n}{\begin{bmatrix}0\\0\\\vdots\\1\end{bmatrix}}$

a "Standard basis" of $\mathbb{R}^n$.

e.g.

$$\begin{bmatrix}1\\2\\3\\4\end{bmatrix} = 1\begin{bmatrix}1\\0\\0\\0\end{bmatrix} + 2\begin{bmatrix}0\\1\\0\\0\end{bmatrix} + 3\begin{bmatrix}0\\0\\1\\0\end{bmatrix} + 4\begin{bmatrix}0\\0\\0\\1\end{bmatrix}.$$

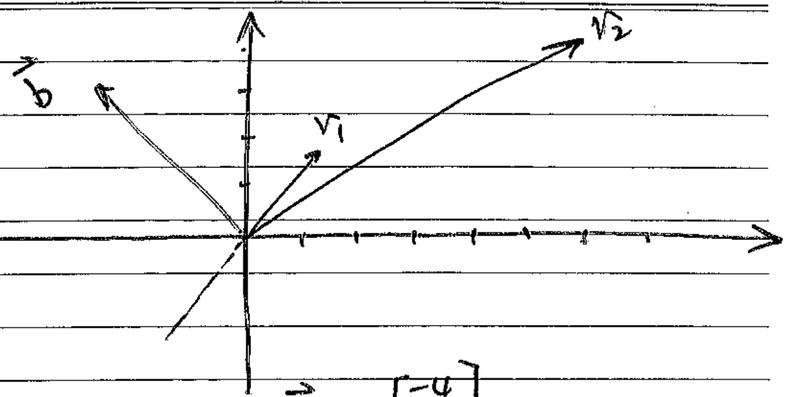
$$= e_1 + 2e_2 + 3e_3 + 4e_4.$$

Check are $e_1, e_2, \dots, e_n$ lin. ind.?

A: Let $A = [e_1 \dots e_4] = \begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}$

So, $Ax = 0$ is just $Ix = 0$.

• $\begin{bmatrix}1\\2\end{bmatrix}, \begin{bmatrix}3\\5\end{bmatrix}$ is a basis of $\mathbb{R}^2$



Alternatively, get $\vec{b} = \begin{bmatrix}-4\\5\end{bmatrix}$.

and solve:

$$\begin{bmatrix}1&4\\2&5\end{bmatrix}\begin{bmatrix}c_1\\c_2\end{bmatrix}=\begin{bmatrix}-4\\5\end{bmatrix}$$

. T/F.

1). $\begin{bmatrix}1\\2\\-4\end{bmatrix}, \begin{bmatrix}7\\-5\\6\end{bmatrix}$ is a basis of $\mathbb{R}^3$?

" failed in the spanning condition "

Does not span $\mathbb{R}^3$

equiv. The linear system

$$\begin{bmatrix}1&7\\2&-5\\-4&6\end{bmatrix}\vec{x}=\vec{b}$$

does not have sol'n for every $\vec{b}$.

example:

$\begin{bmatrix}1\\2\end{bmatrix}, \begin{bmatrix}4\\5\end{bmatrix}, \begin{bmatrix}4\\13\end{bmatrix}$ is a basis $\mathbb{R}^3$?

No, lin. dependence / redundancy in description.

Alternatively,

$$\begin{bmatrix}1&4&4\\2&5&13\end{bmatrix}\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}=\begin{bmatrix}0\\0\end{bmatrix}$$ ★★★.

has a non-trivial sol'n

Properties.

1) Every spanning list can be reduced to a basis by removing redundant elmts.

2). Every lin. ind. list can be extended to a basis.

Def'n The dimension of a V.S. V. is the number of vectors in any basis of V.

Examples:

- dim $\mathbb{R}^n = n$. (think of std. basis).

☆ the rank of a basis

$\Downarrow$

the rank of A is the dim. of the col'n space of A.

Thm: the dim. of col'n space of A agrees with the row space of A.

$$\dim col(A) = \dim row(A)$$ ☆☆☆

Week 4 -3 (TA session)

Rank - Nullity Theorem

$$rank(A) = rank(Null(A)) = n.$$

$$\begin{bmatrix} 2 & 4 & 1 & 0 \\ 1 & 2 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 2 & 4 & 1 & 0 \\ 0 & 6 & 1/2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$2x_1 + 4x_2 + x_3 = 0$$
$$\frac{1}{2}x_3 = 0$$
$$x_3 = 0.$$

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

"zero-vec does not count in the null space."

Example:

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

Date.

No.

Q: find a bases for the null space, row space, column sp. rank.

1). row space

$$A = \left\{ \begin{bmatrix} -2 \\ -1 \\ 0 \end{bmatrix}, \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix}, \begin{bmatrix} 0 \\ -1 \\ -2 \end{bmatrix} \right\}.$$

$$\alpha_1 \begin{bmatrix} -2 \\ -1 \\ 0 \end{bmatrix} + \alpha_2 \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix} + \alpha_3 \begin{bmatrix} 0 \\ -1 \\ -2 \end{bmatrix} = 0.$$

$$\begin{cases} -2\alpha_1 - \alpha_2 = 0 \\ -\alpha_1 \cancel{+} -\alpha_2 - \alpha_3 = 0 \\ -\alpha_2 - 2\alpha_3 = 0 \end{cases}$$

$$\alpha_1 \begin{bmatrix} -2 \\ -1 \\ 0 \end{bmatrix} + \alpha_2 \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix} + \alpha_3 \begin{bmatrix} \cdot \\ , \\ , \end{bmatrix}$$

Symmetric — same

Date.

No.

nullspace $Ax=0$ → row space

For rank: Gaussian Elimination:

$$\begin{bmatrix} -2 & -1 & 0 \\ -1 & -1 & -1 \\ 0 & -1 & -2 \end{bmatrix} \xrightarrow{-\frac{1}{2}} \begin{bmatrix} -2 & -1 & 0 \\ 0 & -\frac{1}{2} & -1 \\ 0 & -1 & -2 \end{bmatrix} -2$$

$$\downarrow$$

$$\begin{bmatrix} -2 & -1 & 0 \\ 0 & -\frac{1}{2} & -1 \\ 0 & 0 & 0 \end{bmatrix}$$

rank = 2

Answer by TA:

$$\begin{cases} -2x_1 - x_2 = 0 \\ -x_1 - x_2 - x_3 = 0 \\ -x_2 - 2x_3 = 0 \end{cases} \rightarrow \begin{matrix} -2x_1 = x_2 \\ x_2 = x_2 \\ x_3 = -\frac{1}{2}x_2 \end{matrix} \rightarrow \begin{bmatrix} -\frac{1}{2} \\ 1 \\ -\frac{1}{2} \end{bmatrix} x_2$$

Date.

No.

S. $\mathcal{B} = \left\{ \begin{bmatrix} -1/2 \\ 1 \\ -1/2 \end{bmatrix} \right\} \leftarrow$ Null(A)

rank(A) = dim(col(A)) + dim(row(A)).

Find the sol'n (if any) to

$A\vec{x} = \vec{b}$ where $\vec{b} = \begin{bmatrix} 0 \\ -1/2 \\ -1 \end{bmatrix}$

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

LU decomposition

$$\begin{bmatrix} 1 & 0 & 0 \\ & 1 & 0 \\ & & 1 \end{bmatrix} \qquad \begin{bmatrix} -2 & -1 & 0 \\ -1 & -1 & -1 \\ 0 & -1 & -2 \end{bmatrix} \Big) -\frac{1}{2}$$

$$\downarrow$$

$$\begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ & & \end{bmatrix} \Big) -\frac{1}{2} \begin{bmatrix} -2 & -1 & 0 \\ 0 & -\frac{1}{2} & -1 \\ 0 & -1 & -2 \end{bmatrix}$$

no sol'n

Date.

No.

Q Prove that $\|Ax\|_2 \le \|A\|_2 \|x\|_2$

First, some reminders.

1). $\|x\|_2 = \sqrt{\sum_{i=1}^{n} |x_i|^2}$

2). $\|A\|_2 = \dfrac{\|Ax\|_2}{\|A\|_2}.$

$$\max_{x \neq 0} \frac{\|Ax\|_2}{\|x\|_2}.$$

Note: This is an "induced" vector, which means it measures how big the size of the output Ax can be w.r.t. the set of the input x.

Pf. we want to show for any vector x, $\|Ax\|_2 \le \|A\|_2 \|x\|_2$

(If $x \neq 0$), rewrite as

$$\frac{\|Ax\|_2}{\|x\|_2} \le \|A\|_2.$$

For any $x \neq 0$, we know that

$$\frac{\|Ax\|_2}{\|x\|_2} \le \max_{x \neq 0} \frac{\|Ax\|_2}{\|x\|_2} = \|A\|_2.$$

So indeed

$$\frac{\|Ax\|_2}{\|x\|_2} \le \|A\|_2$$

or equivalently, $\sim$.

Intuition for conditioning.

Roughly, a matrix is ill-conditioned if it is close to being singular

Example.

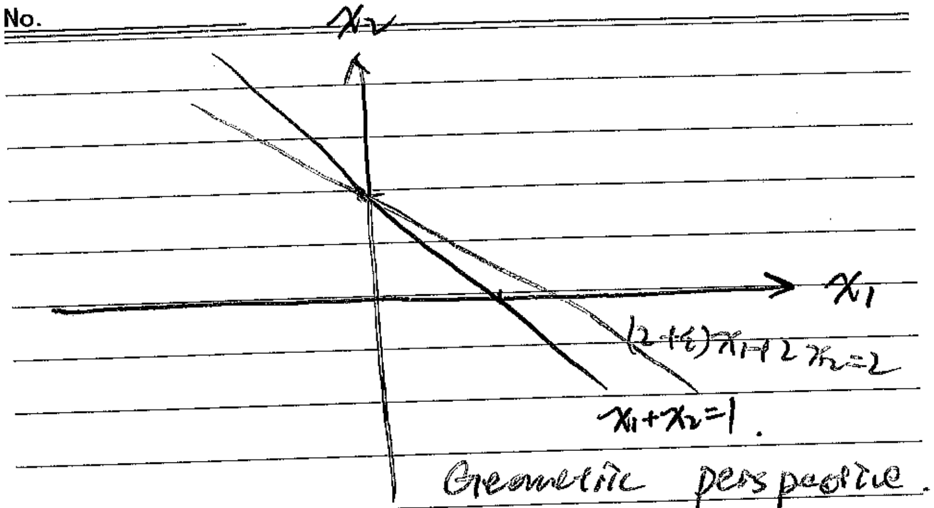
Consider $Ax=b$,

w/ $A = \begin{bmatrix} 1 & 1 \\ 2+\varepsilon & 2 \end{bmatrix}$ & $b = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$.

$\varepsilon \approx 0$ (but $\varepsilon \neq 0$).

$$Ax=b \iff \begin{bmatrix} 1 & 1 \\ 2+\varepsilon & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}.$$

$$\iff x_1 + x_2 = 1.$$
$$(2+\varepsilon)x_1 + 2x_2 = 2$$



Geometric perspective.

⇓

the hyper spaces are almost on top of each other.

⇒ the matrix becomes almost singular.

⇓

the two line become the same line

↓

the matrix does not have a unique sol'n!

Ill-conditioning arises when we done around the boundary of ~~between~~ invertibility & singularity!

Notice two cases.

① $\varepsilon \neq 0$. rows of A are linearly independent.

↓

they form a basis of $\mathbb{R}^2$.

should be a unique sol'n.

② $\varepsilon = 0$. A is just $\begin{bmatrix}1 & 1\\ 2 & 2\end{bmatrix}$.

↓

not linearly indep't

rows are dependent.

LU with row permutations.

Date.

No.

<u>Q</u>: why do we get $PA = LU$?

<u>keypoint</u>: every row operation can be implemented by a mat. mult.

<u>e.g.</u> $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$.

$r_2 \leftarrow r_2 - 4r_1$

$\rightarrow \begin{bmatrix} 1 & 2 & 3 \\ 0 & -3 & -6 \\ 7 & 8 & 9 \end{bmatrix}$ this procedure can be implemented in Gaussian Transform.

$G_1 = \begin{bmatrix} 1 & & \\ -4 & 1 & \\ & & 1 \end{bmatrix}$

first step. $GA = A^{(1)}$ (check).

Can also implement row swaps by multiplying by permutation matrix.

Date.

No.

<u>e.g.</u> Say I want to swap r_2 and r_3 in $A^{(1)}$.

So, $A^{(1)} = \begin{bmatrix} 1 & 2 & 3 \\ 0 & -3 & -6 \\ 7 & 8 & 9 \end{bmatrix}$

$$\downarrow$$

$$\begin{bmatrix} 1 & 2 & 3 \\ 7 & 8 & 9 \\ 0 & -3 & -6 \end{bmatrix} = A^{(2)}.$$

See $P_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$, Then

$$P_1 A^{(1)} = A^{(2)}.$$

So, in total, we've applied:

$$P_1 G A = A^{(2)}$$

To remove $[A^{(2)}]_{21} = 7$, I'll apply

G_2: $G_2 P_1 G A = A^{(3)}$.

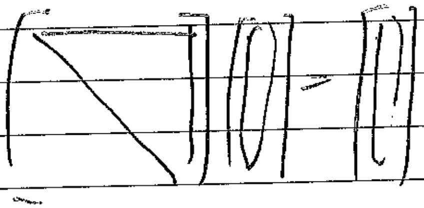
the point is addition operations are angled with permutation operations.

Problem #4

If one uses GE. w/ pivoting. then the resulting system has better conditioning.

Note GE transforms the lin. sys. Ax=b. to an equivalent system, Ux=c that is easy to solve (using back substitution).

b/c U is UT.



Lets try to build a counterexample

Consider earlier example:

$$A = \begin{bmatrix} 1 & 1 \\ 2+\varepsilon & 2 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 2 \end{bmatrix}.$$

Q: What happens if we do GE?

A

$$\left[\begin{array}{cc|c} 1 & 1 & 1 \\ 2+\varepsilon & 2 & 2 \end{array}\right]$$

$$\to \left[\begin{array}{cc|c} 1 & 1 & 1 \\ 0 & -\varepsilon & -\varepsilon \end{array}\right].$$

Date.
No.

Example

$A\begin{bmatrix}0\\0\\1\end{bmatrix}=\vec{0}$.

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

$\det(A)=0$.

Observation: A is not invertible.

Definition The determinant of a square matrix, an $n\times n$ A is given by:

$$\det(A)=\sum_{j=1}^{n}(-1)^{j+1}a_{ij}\det(M_{ij}).$$

where M_{ij} is the sub-matrix obtained from A by removing its i^{th} row + j^{th} col'n.

Note: No need to expand along first row,

Date.
No.

Can in fact use any row you like!

Ex: Using A as in previous example,

$\det(A)=0$.

expand along the second row.

$$\det(A)=0\cdot\det(M_{21})-(1)\cdot\det\left(\begin{bmatrix}2&0\\1&0\end{bmatrix}\right)$$

$$+0\cdot\det.(N_{23}).$$

$$=-\det\left(\begin{bmatrix}2&0\\1&0\end{bmatrix}\right)=0.$$

Properties Vertical bars mean det.

1). Scaling properties e.g. $\det\left(\begin{bmatrix}a_{11}&a_{12}\\a_{21}&a_{22}\end{bmatrix}\right)$.

$$=\begin{vmatrix}a_{11}&a_{12}\\a_{21}&a_{22}\end{vmatrix}.$$

Week 6 -1.

last time { Determinant.
Rank-nullity.

Today { Orthonormal basis: what, why, how?
QR factorization

- - - - - - - - - - - - - - - - - -

Recall: Suppose I want to solve $Ax=b$.

Then: (1): A sol'n exists iff $b \in col(A)$.

(Def'n of mat-vec mult.)

$$Ax=b \iff x_1\vec{a_1} + \cdots + x_n\vec{a_n} = \vec{b}$$

Sol'n iff I can find coeffs to express $\vec{b}$ as lin. comb. of col'ns of A

(2) The null space of A determines the number of sol'ns we have a sol'n for every elt. of $N(A)$.

Q: why?

A: Say x^* is any sol'n.

(So $Ax^* = \vec{b}$) Now take any elt. z in $N(A)$.

(So $Az = \vec{0}$).

↳ observe: $A(\underset{\text{sol'n}}{\underset{\downarrow}{x^*}} + \underset{\text{null space}}{\underset{\downarrow}{z}}) =$

$$Ax^* + \cancel{Az} = \vec{b}$$

So $y = x^* + z$ is also a sol'n to $Ax=b$!

To be precise, the number of free parameters / degrees of freedom in the general sol'n to $A\vec{x} = \vec{b}$ is

$\dim N(A)$, A.K.A nullity of A.

which is $n - rk(A)$.

by RK-Nullity Thm.

Q is what kinds of basis we have for the col'n space.

Q: How do I find these coeffs?

i.e., $x_1, \dots, x_n$, s.t.

$$x_1\vec{a_1} + x_2\vec{a_2} + \dots + x_n\vec{a_n} = \vec{b}.$$

A: Not so easy in general. (use GE/LU fact.) ... But if we have an o.n. basis of col(A), we have a nice explicit formula!

Orthonormal: Orthogonal + normalized.

Orthogonal

Defn The vectors $\vec{v}, \vec{w} \in \mathbb{R}^n$ are orthogonal if $\vec{v}^T\vec{w} = 0$.

Defn: The vectors $q_1, \dots, q_r$ are orthonormal if they are mutually orthogonal and each vec has norm 1.

Explicitly, $q_i^T q_j = \delta_{ij}$. Here

$$\delta_{ij} = \begin{cases} 1, & \text{if } i=j, \\ 0, & \text{if } i\neq j \end{cases}$$

denotes the Kronecker delta.

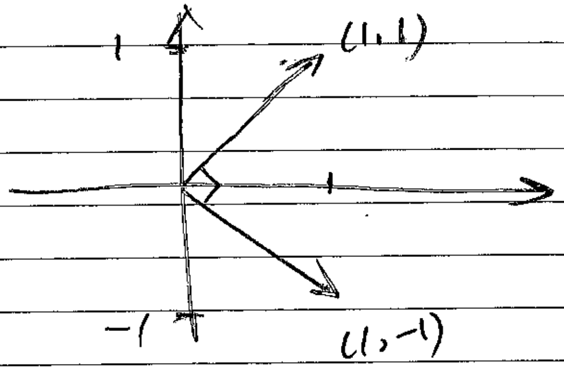
(Recall: $v^Tv = \|v\|_2^2 = \sum_{k=1}^{n} v_k^2$ for any. $v \in \mathbb{R}^n$).

Example: $\begin{bmatrix}1\\-1\end{bmatrix}, \begin{bmatrix}1\\1\end{bmatrix}$ are orthogonal, not orthonormal

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

$$\begin{bmatrix}\frac{1}{2}\\-\frac{1}{2}\end{bmatrix}^T\begin{bmatrix}\frac{1}{2}\\-\frac{1}{2}\end{bmatrix} = \frac{1}{4}+\frac{1}{4} = \frac{1}{2}.$$

observations.



- Orthogonality : → Synoymous w/ perpendicular in $\mathbb{R}^2$ & $\mathbb{R}^3$
- Notion of "right angles" in general V.S.

Q: why should we care about these basis?

A: Suppose (for simplicity).

A is $n\times n$ non-singular and we want to solve $Ax=b$.

If $q_1, \dots, q_n$ is an o.n. basis of col(A), then I can readily compute $c_1, \dots, c_n$, s.t.

$$c_1\vec{q}_1 + c_2\vec{q}_2 + \dots + c_n\vec{q}_n = \vec{b}.$$

Q: How?

A: Use $\vec{q}_i^T\vec{q}_j = \delta_{ij}$! Multiply by $\vec{q}_i^T$ to obtain

Date.

No.

$$c_1\cancelto{0}{(\vec{q}_i^T\vec{q}_1)} + \cdots + c_i\cancelto{1}{(\vec{q}_i^T\vec{q}_i)} +$$

$$\cdots + c_n\cancelto{0}{(\vec{q}_i^T\vec{q}_n)} = (\vec{q}_i^T\vec{b}).$$

$$\Rightarrow \vec{c}_i \cdot 1 = \vec{q}_i^T\vec{b}$$

$\Leftrightarrow \vec{c}_i = \vec{q}_i^T\vec{b}$ ...> Success =)

Given an o.n. basis for col(A), I can find $c_1, \dots, c_n$, s.t. $\vec{b} = c_1\vec{q}_1 + \cdots + c_n\vec{q}_n$ using $c_i = \vec{q}_i\vec{b}$

Matrix perspective.

Finding $c_1, \dots, c_n$ is equiv. to $c_1\vec{q}_1 + \cdots + c_n\vec{q}_n = \vec{b}$ $\Leftrightarrow$.

$$\begin{bmatrix} q_1 & \cdots & q_n \end{bmatrix}\begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix} = \vec{b}$$

Date.

No.

$$\Leftrightarrow Q\vec{c} = \vec{b}$$

Note: Much like if coeff. mat. is lower/upper triangular, lin. sys is easy to solve. ($\mathcal{O}(n^2)$ cost) if mat. has o.n. colns.

In particular, sol'n:

$$\vec{c} = \begin{bmatrix} \vec{q}_1^T b \\ \vdots \\ \vec{q}_n^T b \end{bmatrix} = \begin{bmatrix} - \vec{q}_1^T - \\ - \vec{q}_2^T - \\ \vdots \\ - \vec{q}_n^T - \end{bmatrix}\vec{b}$$

$$= Q^T\vec{b}$$

Let's take a closer look... Multiplying $Q\vec{c} = b$ by Q^T gives

$$(Q^TQ)\vec{c} = Q^T\vec{b}$$

Since Q is invertible b/c orthogonality $\Rightarrow$ lin. ind., we must have: $\underset{\downarrow}{Q^TQ=I}$

Notice: this is equivalent to

$$q_i^Tq_j = \delta_{ij}$$

$$\begin{bmatrix} q_i^Tq_j & - \\ & \end{bmatrix} = \begin{bmatrix} — q_i^T — \end{bmatrix}\begin{bmatrix} q_j & \cdots \end{bmatrix}$$

Defn: An $n\times n$ matrix is orthogonal if its col'ns are $\underbrace{\text{orthonormal}}_{\downarrow \text{ o.n.s}}$.

Equivalently, $n\times n$ Q is orthogonal

iff $Q^TQ=I$

Note: if Q is orthogonal, $Q^{-1}=Q^T$

Example: $H = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$, is orthogonal.

(Hadamard gate)

Q: How can we produce an o.n. basis for col(A)?

A. Use standard Gram-schimidt

Caveat: No great for numerics.

Basic Idea: Step-by-Step

Construction: at each step, use a new col'n of A and remove components parallel to (contained in span of) vectors you've already found!

Say A is $n \times n$.

Goal: Construct o.n. list $q_1, \dots, q_n$, s.t. $\text{span}(q_1, \dots, q_k) = \text{span}(\vec{a}_1, \dots, \vec{a}_k)$ for $k = 1, \dots, n$.

Gram-Schmidt Orthogonalization:

Step 1: Take $\vec{q}_1 = \frac{\vec{a}_1}{\|a_1\|}$.

(simply normalize).

Step 2: want o.n. $\vec{q}_1, \vec{q}_2$, s.t.

$\vec{a}_2 \in \text{span}(\vec{q}_1, \vec{q}_2)$.

So I need:

$$c_1 \vec{q}_1 + c_2 \vec{q}_2 = \vec{a}_2 .$$

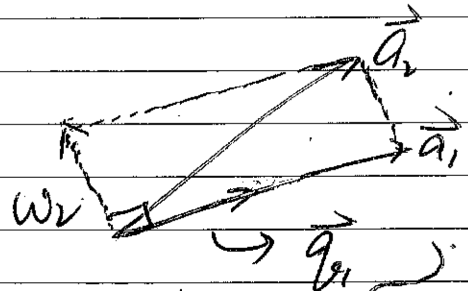
Before: $c_1 = \vec{q}_1^T \vec{a}_2$, so.

$$c_2 \vec{q}_2 = \underbrace{\vec{a}_2 - (\vec{q}_1^T \vec{a}_2)\vec{q}_1}.$$

So $\vec{q}_2$ as normalized

$$\vec{w}_2 = \vec{a}_2 - (\vec{q}_1^T \vec{a}_2)\vec{q}_1$$

$$\therefore \vec{q}_2 = \frac{\vec{w}_2}{\|\vec{w}_2\|}$$



$a_2 - \text{proj}_{\vec{q}_1}(\vec{a}_2)$.

$\text{proj}_{\vec{q}_1}(a_2) = (\vec{q}_1^T a_2)\vec{q}_1$

Step 3 q_1, q_2, q_3

s.t. $\vec{a_3} \in \text{span}(q_1, q_2, q_3)$.

So we want

$c_1\vec{q_1} + c_2\vec{q_2} + c_3\vec{q_3} = \vec{a_3}$

From before:

$c_1 = q_1^T \vec{a_3}$,

$c_2 = q_2^T \vec{a_3}$.

So: $c_3\vec{q_3} = \vec{a_3} - (q_1^T\vec{a_3})\, q_1 - (q_2^T\vec{a_3})\, q_2$

is orthogonal to $\vec{q_1}, \vec{q_2}$.

So take $\vec{q_3} = \dfrac{\vec{w_3}}{\|w_3\|}$

Step k: Remove from $\vec{a_k}$ components

parallel to $\vec{q_1}, \dots, \vec{q_{k-1}}$ and normalize!

$$w_k = \vec{a_k} - (q_1^T \vec{a_k})\, q_1 - \dots - (q_{k-1}^T \vec{a_k})\, q_{k-1}.$$

$$\vec{q_k} = \frac{w_k}{\|w_k\|}. \quad \text{(normalize!)}$$

Notice: We obtain $A = QR$, w/ Q orthogonal and R upper triangular if we keep track of $r_{ik} = q_i^T a_k$!

$$[\vec{a_1} \ \cdots \ \vec{a_n}] = A = Q \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ & r_{22} & r_{23} \\ & & r_{33} \end{pmatrix}$$

Look at the second col'n,

$$\vec{a_2} = Q \begin{bmatrix} r_{12} \\ r_{22} \\ 0 \end{bmatrix}$$

$$= r_{12}\, q_1 + r_{22}\, \vec{q_2}$$

Date.
No.

Week 7 - 1

$A\vec{x} = \vec{b}$.

- Gauss Elimination
- QR Decomposition
- Iterative Methods.

Regression Models.

Find solutions minimize:

$$\vec{\varepsilon} = \vec{y} - A\vec{t}$$

Week 7 - 2.

Date.
No.

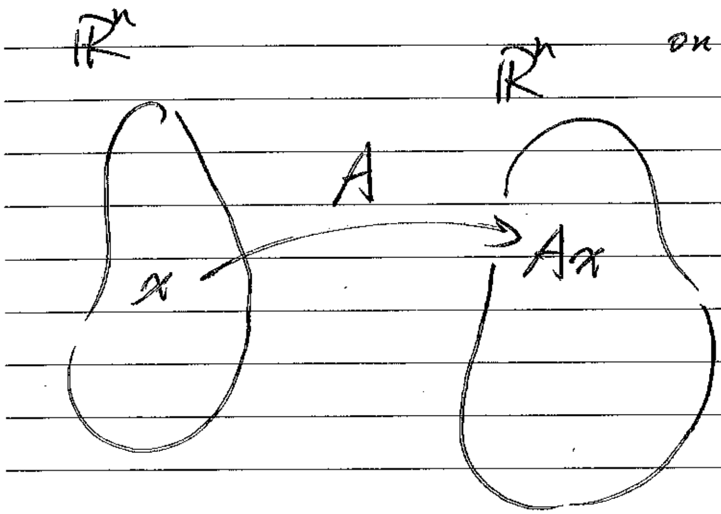
Square matrices

↳ represents operators

⇓

linear transformation

on $\mathbb{R}^n$.



There are "special" vectors that don't change direction under the transformation.

In particular, given $n \times n$ A,

an eigenvector is $v \in \mathbb{R}^n$ s.t.

$$A\vec{v} \sim \vec{v}$$

↳ equivalent to saying

there's a scalar λ

s.t. $A\vec{v} = \lambda\vec{v}$

$\downarrow$

Def'n eigenvalue.

Given an $n \times n$ A, the scalar λ

is an eigenvalue of A if there

is $\vec{v} \neq 0$ s.t.

$$A\vec{v} = \lambda\vec{v}$$

The vec $\vec{v}$ is a λ-eigenvector

of A.

* square matrices don't change

the dimension of the vector.

Q: How to find eigenvectors &

determine eigenvalues?

A: From $A\vec{v} = \lambda\vec{v}$.

$$\Leftrightarrow A\vec{v} - \lambda\vec{v} = 0$$

$$\downarrow$$

$$\lambda I.$$

$$\Leftrightarrow (A - \lambda I)\vec{v} = 0$$

TBC →

$$\phi = (u_1 - u_0) + P_0(\cancel{V_1} - \cancel{V_0}) - T_0(S_1 - S_0).$$

$PV = ZRT$ ↑

$$= \underline{(h_1 - h_0)} - R(Z_1T_1 - Z_0T_0) + P_0(V_1 - V_0)$$

$$\downarrow \quad \downarrow \quad - \underline{T_0(S_1 - S_0)}$$

Confirm: $C_p(T_1 - T_0)$

$$(h_1 - h_0) = \Delta h_{ideal} - \underset{\text{residual}}{(h_1^R - h_0^R)}$$

$$(S_1 - S_0) = -(S_1^R - S_0^R) + \Delta S_{ideal}$$

$$h_1 = \qquad \left\{ C_p \ln\frac{T_1}{T_0} - R\ln\frac{P_1}{P_0} \right\}$$

'R' these one from ~~the~~ table ..?

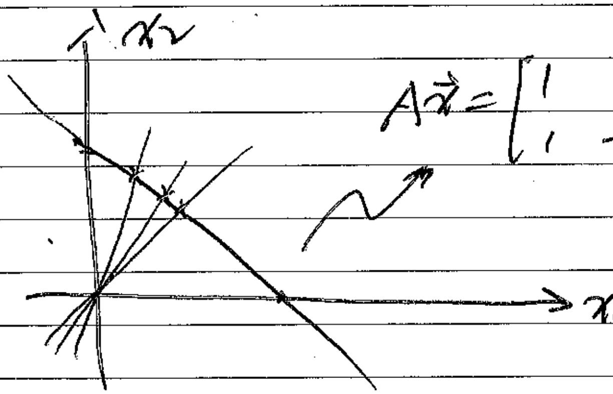
Week 6-2.

Sol'n of linear systems.

$$A\vec{x} = \vec{b}.$$

approximate sol'n $\vec{x}' \simeq \vec{x}$

- limited precision
- Computing w/ uncertainty.



$$A\vec{x} = \begin{bmatrix} 1 & 2 \\ 1 & -1\pm 0.1 \end{bmatrix} \vec{x} = \begin{bmatrix} 3 \\ 0 \end{bmatrix}.$$

$$a_{22} = -1 \rightarrow \vec{x} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$A\vec{x} = \vec{b}$$

$$\lim_{k\to\infty} \vec{x}^{(k)} = \vec{x}$$

- guess $\vec{x}^{(0)}$
- Sequence: $\vec{x}^{(0)} \longrightarrow \vec{x}^{(1)} \rightarrow \dots \rightarrow \vec{x}^{(k)}$

Sequence terminates: $\|\vec{x}^{(k)} - \vec{x}\| < tol.$

↳ simple, less computational cost

☒ Definitions:

Error: $\vec{e}^{(k)} = \vec{x}^{(k)} - \vec{x}$

Residual: $\vec{r}^{(k)} = \vec{b} - A\vec{x}^{(k)}$

approach converge, error & residual

→ 0: $\lim_{k\to\infty}(x^{(k)} - x) = \lim_{k\to\infty}(A\vec{x}^{(k)} - \vec{b}).$

In terms of norms. → ...

Splitting $\vec{A} = M - N$

$A\vec{x} = \vec{b} = (M - N)\vec{x} = \vec{b}$

$A = M - N$ → decomposition

Iterative process: $+b$

$M\vec{x}^{(1)} = N\vec{x}^{(0)} + b_1$

$Mx^{(2)} = N\vec{x}^{(2)} + b$

$\vdots$

$Mx^{(k)} = N\vec{x}^{(k-1)} - b$

$(N - M = A$

Simplest Split system

$\|e^{(k)}\| = \|e^{(k)0} G^{k}\|$

$= \|G e^{(0)}\| \geq 0$

$\longrightarrow C G > 0.$

Converge condition.

Jacobi Solver. →

↓

special radius → test convergence of norm.

Jacobi:

- $M = D$.
- $N = -L(A) - U(A)$

Alternative View → Lecture.

TBC

Solve $(A - \lambda I)\vec{v} = 0$

Keypoint. Only possible if

$A - \lambda I$ is singular,

$\Leftrightarrow$ cols are dependent

$\Leftrightarrow$ $N(A) \neq \{0\}$.

$\Leftrightarrow$ $\det(A - \lambda I) = 0$

Keypoint We can only find eigenvectors corresponding to eigenvalues → Can only solve.

$(A - \lambda I)\vec{v} = 0$. iff $\det(A - \lambda I) = 0$

lets tackle the latter

first Example:

let $A=\begin{bmatrix}0&1\\1&0\end{bmatrix}$. Find λ s.t.

$$\det(A-\lambda I)=0.$$

A: ① compute determinant.

$$\begin{vmatrix}-\lambda & 1\\ 1 & -\lambda\end{vmatrix} = \lambda^2-1$$

↓

② solve polynomial: $\lambda=\pm1$.

=> find eigenvalues

③ we can now solve for

eigenvectors.

e.g. find eigenvecs corresponding to $\lambda=1$, by solving

$$(A-I)\vec{v}=0$$

$$\Rightarrow \begin{bmatrix}-1&1\\1&-1\end{bmatrix}\vec{v}=0$$

$\lambda=1$ → the matrix has a nullspace, arrows parallel to each other or scalable to each other.

Say $B=A-I=\begin{bmatrix}-1&1\\1&-1\end{bmatrix}$.

$[\vec{b}_1\ \vec{b}_2] \Rightarrow \vec{b}_1=-\vec{b}_2$

$\iff \vec{b_1} + \vec{b_2} = 0$

$\iff [\vec{b_1}\ \vec{b_2}]\begin{bmatrix}1\\1\end{bmatrix} = 0$

So $v = \begin{bmatrix}1\\1\end{bmatrix}$ is a 1-eigenvec of A

Now repeat process to find an eigenvec corresponding to $\lambda = -1$

In this case, solve

$$(A+I)\vec{v} = 0.$$

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

$\vec{v} = \begin{bmatrix}1\\-1\end{bmatrix}$ is an eigenvector of A.

Observations:

→ For any A; and any eigenvalue λ of A, there is more than one λ-eigenvector.

... why this is true?

Notice: Any linear combination of λ eigenvectors is also a λ-eigenvector. (i.e. the eigenspace)

Say $\vec{v_1}$ & $\vec{v_2}$ are λ-eigenvectors.

Notice $A(c_1\vec{v_1} + c_2\vec{v_2}) =$

$= c_1(A\vec{v_1}) + c_2(A\vec{v_2})$. (linear trans)

Date.

No.

$= c_1\lambda_1\vec{v_1} + c_2\lambda_2\vec{v_2}$

$= \lambda\underbrace{(c_1\vec{v_1} + c_2\vec{v_2})}$

∴ another eigen-vector!!!

$\vec{v} = c_1\vec{v_1} + c_2\vec{v_2}$ is also a λ- eigenvector.

bc/ $A\vec{v} = \lambda\vec{v}$.

Q: Given an eigenvalue λ of A, what's the dimension of the subspace of the corresponding eigenvector?

A: the number of linearly ind. λ- eigenvectors is $\dim[N(A-\lambda I)]$

Date.

No.

Example

Consider $I = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$ &

$J = \begin{bmatrix}1 & 1\\ 0 & 1\end{bmatrix}$

Notice $I\vec{v} = (1)\cdot\vec{v}$ &

$J\begin{bmatrix}1\\0\end{bmatrix} = (1)\begin{bmatrix}1\\0\end{bmatrix}$. So

1 is an eigenvalue for I & J.

BUT how many lin. ind. eigenvecs?

① $(I - I)\vec{v} = 0$

$\downarrow$

So we have 2 lin. ind. eigenvecs.

② $(J - I)\vec{v} = 0 \rightarrow \begin{bmatrix} & 1\\ & \end{bmatrix}\vec{v} = 0$.

$\downarrow$

$\dim[N(J-I)] = 2 - \text{rank} = 1$ rank $= 1$.

So we just have 1 lin. ind eigenvec.

... #Jordan Blocks.

Q: Do eigenvals exist?

How many are there?

A:

Key point: $\det(A-\lambda I)$ is a degree n polynomial.

So it always has n solutions.

↓

don't have to be distinct, & don't have to be real.

1) T/F: A is singular IFF 0 is an eigenvalue of A.

True: If A is singular, then there is $\vec{v} \neq 0$, s.t. $A\vec{v} = \vec{0} = 0 \cdot \vec{v}$

In other words, $\vec{v} \in N(A) \Rightarrow \vec{v}$ is a 0-eigenvector of A.

Alternatively, we know that

$0 = \det(A) = \det(A - 0 \cdot I)$ so

$\det(A - \lambda I) = 0$ when $\lambda = 0$.

Conversely, if 0 is an eigenvalue, then there is a corresponding eigenvector, i.e., $\vec{v} \neq 0$, s.t.

$A\vec{v}=0. \vec{v}=0.$

So if 0 is an eigenval, every corresponding eigen vec $\vec{v} \in N(A)$.

Alternatively, if $\lambda=0$ is an eigenvalue, $\det(A-\lambda I)=0$, when $\lambda=0$, which means $\det(A)=0$.

2). T/F A & A^T have the same eigenval.

True Proof: Eigenvalues are roots of $\det(A-\lambda I)$; we know $\det(B)=\det(B^T)$ for any B.

So. $\det(A^T-\lambda I)=\det(A^T-\lambda I^T)$

$$=\det(A-\lambda I)^T$$

$$=\det(A-\lambda I).$$

Recall: $\det(A-B) \neq \det(A) \mp \det(B)$.

in general!

But: $\begin{vmatrix} 1 & 2+a & 3 \\ 4 & 5+b & 6 \\ 7 & 8+c & 9 \end{vmatrix} = \begin{vmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{vmatrix}$

$+ \begin{vmatrix} 1 & a & 3 \\ 4 & b & 6 \\ 7 & c & 9 \end{vmatrix}$

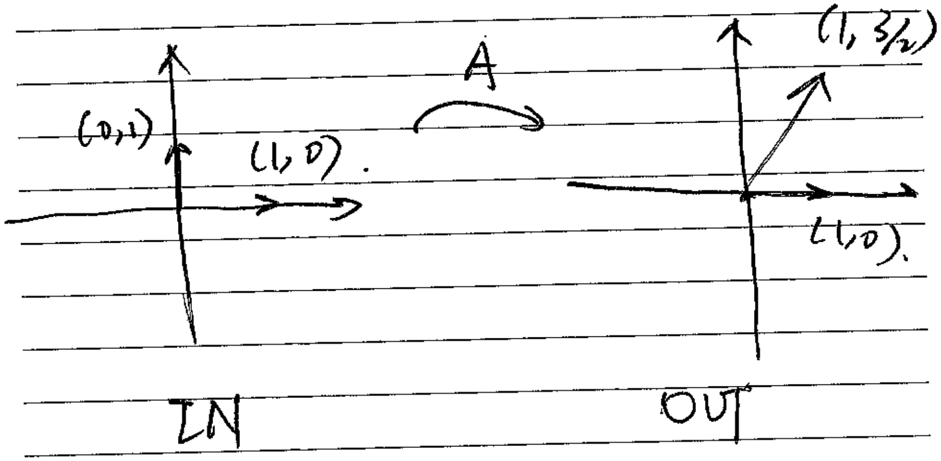
Operator Perspective

A 2x2 matrix transforms vectors in the xy-plane.

Example: $A=\begin{bmatrix}1 & 1\\ 0 & 3/2\end{bmatrix}$

$$\vec{x} \xrightarrow{f} A\vec{x}$$

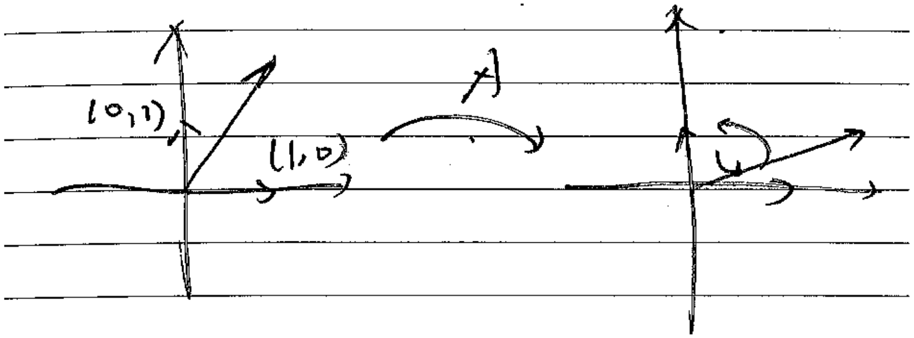
$$f(\vec{x}) = A\vec{x}$$



From last lecture, $A=\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}$,

we found eigenvalues $\lambda=-1, 1$

w/ eigenvectors $\begin{bmatrix}-1\\ 1\end{bmatrix}$ and $\begin{bmatrix}1\\ 1\end{bmatrix}$



$(1,0) \to (0,1)$

$(0,1) \to (1,0)$

$(1,2) \to (2,1)$

Note: A is a householder reflection!

$$A_v = I - \frac{2}{v^Tv} vv^T$$

$$v = \begin{bmatrix}1\\ 1\end{bmatrix}$$

Recall: we said the characteristic polynomial $\chi(z) = \det(A - zI)$

has degree n for $n\times n$ A. So A has n eigenvalues.

Let's begin with easy/good case.

Case: $\chi_A(z)$ has n distinct real roots.

Let's count lin. ind. eigenvecs.

① For each eigenvalue, there must be an eigen vec.

(But if $\det(A-\lambda I)=0$ then $A-\lambda I$ is singular, so there is $\vec{v}\neq 0$, s.t. $(A-\lambda I)\vec{v}=0$.

$\Leftrightarrow A\vec{v}=\lambda\vec{v}$

② Eigenvectors corresponding to distinct

eigenvalues are lin. ind.

Q: why?

A: (Sketch). Suppose $v_1, \dots, v_k$ are eigenvecs of A w/ corresponding eigenvalues. $\lambda_1, \dots, \lambda_k$ (distinct),

If $c_1\vec{v}_1 + \cdots + c_k\vec{v}_k = 0$ (*) we need to show $c_1 = \cdots = c_k = 0$

Hint: Multiply (*) by $(A-\lambda_j I)$

☆☆☆

So, if A has n distinct eigenvals, it has n lin. ind eigenvectors!

→ We have $\underbrace{\text{a basis of } \mathbb{R}^n}$.

named as "eigenbasis"

Concretely, we have

$$Av_j = \lambda_j v_j \text{ for } j = 1, \dots, k.$$

and $v_1, \dots, v_n$ are lin. ind.

Let's combine these into a single matrix relation.

$$[A\vec{v}_1 \ \ A\vec{v}_2 \ \dots \ A\vec{v}_n] = [\lambda_1\vec{v}_1 \ \dots \ \lambda_n\vec{v}_n]$$

$$A \begin{bmatrix} | & | & & | \\ \vec{v}_1 & \vec{v}_2 & \cdots & \vec{v}_n \\ | & | & & | \end{bmatrix} = \begin{bmatrix} | & | & & | \\ \lambda_1\vec{v}_1 & \lambda_2\vec{v}_2 & \cdots & \lambda_n\vec{v}_n \\ | & | & & | \end{bmatrix}$$

$$\begin{bmatrix} 2 & \\ & 3 \end{bmatrix}\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 2 & 4 \\ 9 & 12 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}\begin{bmatrix} 2 & \\ & 3 \end{bmatrix} = \begin{bmatrix} 2 & 6 \\ 6 & 12 \end{bmatrix}.$$

So

$$A \overbrace{\begin{bmatrix} | & | & & | \\ \vec{v}_1 & \vec{v}_2 & \cdots & \vec{v}_n \\ | & | & & | \end{bmatrix}}^{V} = \begin{bmatrix} | & | & & | \\ \vec{v}_1 & \vec{v}_2 & \cdots & \vec{v}_n \\ | & | & & | \end{bmatrix} \overbrace{\begin{bmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \ddots \ \lambda_n \end{bmatrix}}^{\Lambda}$$

$$\iff AV = V\Lambda$$

Q: is V invertible?

A: Yes, b/c eigenvalues are lin. ind.

$$A = V\Lambda V^{-1}$$

★★★ the canonical form. //

Note: Distinct eigenvals is sufficient for existence of decomp., but NOT necessary. What we really need is n lin. ind eigenvecs.

Degenerate Cases: (repeated eigenval)

Example: Repeated eigenvals but diagonalizable.

$$I = \begin{bmatrix} 1 & \\ & 1 \end{bmatrix}$$

Eigenvals 1, 1

Eigenvecs. 2 lin. ind.

eigenvecs: $\vec{v_1} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $\vec{v_2} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$.

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & \\ & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^{-1}.$$

Repeated eigenvals but NOT diagonalizable

$$J = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix},$$

Eigenvals: $\chi_J(z) = (z-1)^2$

1, 1,

Eigenvecs Only $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$, Not enough lin. ind eigenvecs to form eigenbasis.

Def'n: Algebraic multi. of eigenval λ

of A is the multi. of λ as a root of char. poly.

Def'n Geom multi is $\dim N(A-\lambda I)$

$= \#$ of lin. ind. λ-eigenvecs.

Diagonalizability $\Leftrightarrow$ alg. and geom. multi. coincide for every eigenval. ("enough lin ind. eigenvecs for each eigenval.")

Note: Even real matrs can have complex eigenvals.

Example: $R = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$.

$\chi_R(z) = z^2+1$.

→ Eigenvecs are also complex.

How is this useful in science/eng.?

Systems of ODE.

$$\vec{y}'(t) = \begin{bmatrix} y_1'(t) \\ \vdots \\ y_n'(t) \end{bmatrix} = Ay(t).$$

e.g. if $n=1$,

$$y' = ay.$$

Exponential growth/decay ...

i.e. $y(t) = \overset{\rightsquigarrow y^{(0)}}{K} e^{at}$.

where $n > 1$, try writing

$$\vec{y}(t) = e^{At} y(0)$$

Here, set $e^{At} = \sum_{n=0}^{\infty} \frac{(At)^n}{n!}$

#Spoiler Canonical form makes this computation simple!

key points

least-square / norm sol'n arise when $A\vec{x} = \vec{b}$. has no sol'n or too many.

Least-Squares

Approximate sol'n when $A\vec{x} = \vec{b}$ is not consistent.

Typically, most element interesting for over-determined systems (too many rows / constraints, so there is no way to meet all requirements exactly).

Equivalently, $\vec{b} \notin \overset{\operatorname{col}(A)}{\overset{/\!/}{\operatorname{span}(\vec{a}_1, \dots, \vec{a}_n)}}$.

So we ask for the next best thing, i.e., find $\vec{x} \in \mathbb{R}^n$ s.t.

$$\underset{\vec{x}}{\text{minimize}} \ \|A\vec{x}-\vec{b}\|.$$

Let $\vec{r}(\vec{x}) = A\vec{x} - \vec{b}$

Goal. make residual $\vec{r}$ as small as possible.

Q: How to find such an $\vec{x}$?

A: One option, Calculus!

i.e. let $l(\vec{x}) = \|\vec{r}(\vec{x})\|_2$

and solve $\nabla \tilde{l} = 0$.

Good news. If we square the loss, we can find $\vec{x}$ by solving a linear system!

Normal equations

$$l(\vec{x}) = \|r(x)\|_2^2 = (A\vec{x}-\vec{b})^T(A\vec{x}-\vec{b})$$

$$= \vec{x}^T A^T A \vec{x} - 2b^T A \vec{x} + b^T b$$

So, $\nabla l(\vec{x}) = 0$ is equiv.

$$A^T A \vec{x} = A^T \vec{b} \quad \text{(normal eqn.)}$$

↑ will have a soln as long as the colns are INDEPENDENT!!!

↓

Even if soln $A\vec{x}=\vec{b}$ does not exist!

Date.

No.

Q: why does A need to have full rank (n. lin. ind. colms $\Updownarrow$ $\mathcal{N}(A)=0$).

A: Fact. $\mathcal{N}(A^TA) = \mathcal{N}(A)$ for any A, → ...prove this

And $\mathcal{N}(A^TA)=0$ means A^TA is invertible

So $\vec{x}_{LS} = (A^TA)^{-1}A^T\vec{b}$

Notes about numerics

- The condition number:

$$K(A^TA) = K(A)^2.$$

of the coeff mat for normal

Date.

No.

eqns can be much bigger than that of A!!!

Fix. Use QR fact.

Practice Exam Pb. 2

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

↳ A full rank ✓

... ***

Notice if $B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$.

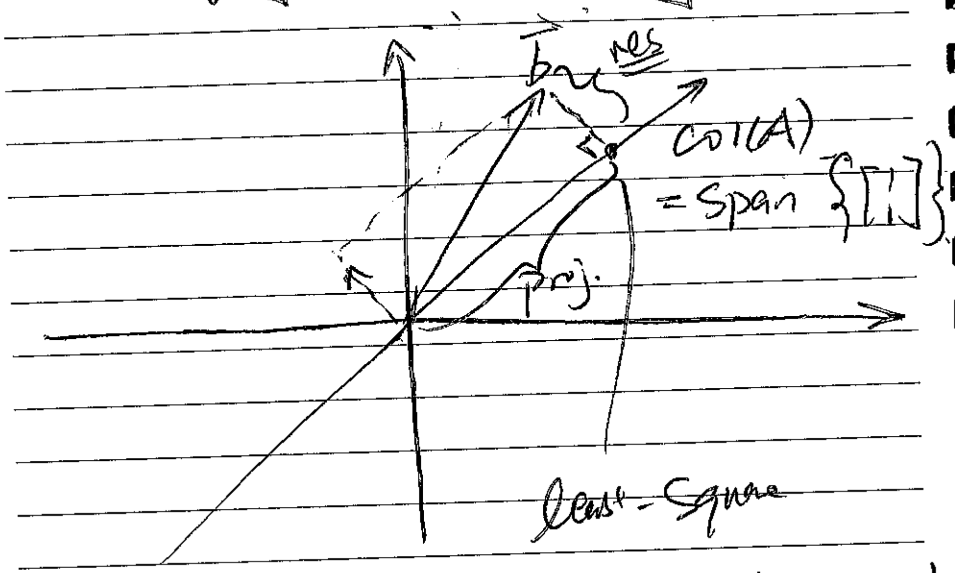
$$B^{-1} = \frac{1}{\det(B)} \cdot \begin{bmatrix} b_{22} & -b_{12} \\ -b_{21} & b_{11} \end{bmatrix}.$$

$$\boxed{\vec{x}_{LS} = (A^TA)^{-1}A^T\vec{b}}$$

→ Picture time ... $A\vec{x}=\vec{b}$

Suppose I want to solve ↗.

$A=\begin{bmatrix}1\\1\end{bmatrix}$, $b=\begin{bmatrix}1\\2\end{bmatrix}$. $x\in\mathbb{R}$


$\vec{b}$
res
col(A)
$=\text{Span}\left\{\begin{bmatrix}1\\1\end{bmatrix}\right\}$
proj.

least-square soln ... (closest) to $\vec{b}$ in col(A).

That's why $\vec{e}=\vec{r}(x_{LS})$ is $\perp$ to col(A), i.e., $A^T\vec{e}=\vec{0}$.

verify:

$$x_{LS} = (A^TA)^{-1}A^T\vec{b}$$

$$= ([2])^{-1}[1\ \ 1]\begin{bmatrix}1\\2\end{bmatrix}$$

$$= 3/2$$

check $\vec{e} = Ax_{LS}-\vec{b}$

$$= (3/2)\begin{bmatrix}1\\1\end{bmatrix}-\begin{bmatrix}1\\2\end{bmatrix}$$

$$= 1/2\begin{bmatrix}1\\-1\end{bmatrix}.$$

least norm sol'n

Date.
No.

Typically most. redundant for undetermined when we want to select a certain soln amongst infinitely many.

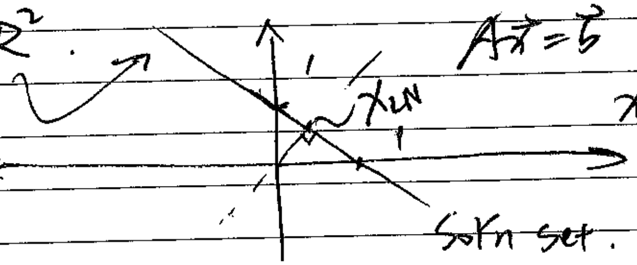
$$\min_{\vec{x}} \|\vec{x}\|_2 .$$
$$\text{s.t.} \quad A\vec{x} = \vec{b} .$$

Least norm

Suppose we want to solve LN- soln to $A\vec{x}=\vec{b}$,

$A = [1, 1]$ $\quad \vec{b} = [1]$.

$x \in \mathbb{R}^2$. $\quad A\vec{x} = \vec{b} \iff$. $\quad x_1 + x_2 = 1$

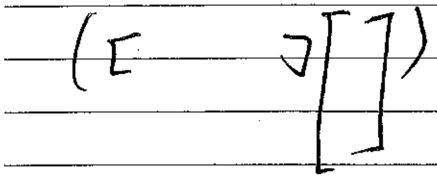


Date.
No.

↳ amongst all possible solns, chose one closest to the origin.

Procedure:

① Use (right) pseudoinverse



$x_{LN} = A^T (AA^T)^{-1} b.$

~~***~~
Iterative metho
prob.1 -HW

$\boxed{T/F}$

② First find general soln, (use calculus!) e.g..

$$\vec{x}(t) = \begin{bmatrix} 1-t \\ t \end{bmatrix}, \quad \text{compute}$$

norm, $\|x(t)\|^2 = (1-t)^2 + t$

and minimize!

Date.

No.

$A \rightarrow \lambda$

what are the eigenvals of

A^{-1}, A^{T}, ... related to A

square invertible ?

A

↑ inv.

$\det(A) = 0$

Singularity ~~ex~~*.

Week 9 - 1.

Date.

No.

- power method
- eigenval, det., norm
- spectral representation

→ Review on prev. concepts.

rank - Gauss diagonalization.

↑ linear combination.

What's left? power method, QR iter., Singular value decomposition.

Now. computing the eigenval.

$\det(A - \lambda I) = 0$

↖ can only used for small n.

$|\lambda_{min}| > 0$

$\kappa(A) = |\lambda_{max}| / |\lambda_{min}|$.

$|\lambda_{min}(G)| < 1 \rightarrow \|G^k\| < 1$.

Power method.

diagonalizable $A \in \mathbb{R}^{n\times n}$.

$|\lambda_1| > |\lambda_2| > \cdots > |\lambda_n|$.

iterate: Given $\vec{u}^{(0)}$.

$\vec{u}^{(1)} = A\vec{u}^{(0)}$.

$\vec{u}^{(2)} = A\vec{u}^{(1)} = A^2\vec{u}^{(0)}$.

$\vdots$

$\vec{u}^{(k)} = A\vec{u}^{(k-1)} = A^k\vec{u}^{(0)}$.

Project $\vec{u}^{(0)}$ on the basis:

$$\vec{u}^{(0)} = \sum_{i=1}^{n} \alpha_i \vec{v}_i$$

$$\rightarrow \vec{u}^{(1)} = A\vec{u}^{(0)} = A\sum_{i=1}^{n} \alpha_i \vec{v}_i$$

$$= \sum_{i=1}^{n} \alpha_i \lambda_i \vec{v}_i$$

$$\vec{u}^{(k)} = A^k \vec{u}^{(0)} = \sum_{i=1}^{n} \alpha_i \lambda_i^k \vec{v}_i$$

$$= \lambda_1^k \left[\alpha_1 \vec{v}_1 + \sum_{i=2}^{n} \alpha_i^k \left(\frac{\lambda_i}{\lambda_1}\right)^k \vec{v}_i \right].$$

Recall: $|\lambda_1| > \cdots > |\lambda_n|$

$\rightarrow \vec{u}^{(k)} \simeq \lambda_1^k \alpha_1 \vec{v}_1$

$\vec{u}^{(k)}$ aligns towards $\vec{v}_1$.

- diagonalizable matrix $A \in \mathbb{R}^{n\times n}$.

$$\vec{u}^{(k)} = A\vec{u}^{(k-1)} = A^k \vec{u}^{(0)}.$$

$$A\vec{v}_1 = \lambda_1 \vec{v}_1 \rightarrow \vec{v}_1^T A \vec{v}_1 = \vec{v}_1^T \lambda_1 \vec{v}_1.$$

$$\lambda_1 = \frac{\vec{v}_1^T A \vec{v}_1}{\vec{v}_1^T \vec{v}_1}.$$

Actual power iterations: normalization,

- $\vec{u}^{(k)} = A\vec{u}^{(k-1)}$
- $\vec{w}^{(k)} = \vec{u}^{(k)} / \|\vec{u}^{(k)}\|$.
- $\lambda^{(k)} = (\vec{w}^{(k)})^T A \vec{w}^{(k)}$.

... k.

Convergence controlled by eigenval getting smaller

↓

λ_1 sufficiently larger than λ_2

get $\lambda_{min} \rightarrow$ power method for A^{-1}

not λ_{max} or λ_{min}

$A - \mu I \Longleftrightarrow \lambda - \mu$.

$$(A-\mu I)\vec{v} = (\lambda-\mu)\vec{v}.$$

↓

$\hat{A} = A - \mu I$, $\hat{\lambda} = \lambda - \mu$

$$\hat{A}\vec{v} = \hat{\lambda}\vec{v}.$$

largest $\frac{1}{\hat{\lambda}} \rightarrow \frac{1}{\lambda-\mu}$.

... Shifted Inverse Power Iterations.

▶ $A \in \mathbb{R}^{n\times n}$ ← power iter.

↓

converges to the spectral radius λ_{max}.

convergence speed depend on λ_2/λ_1.

$|\lambda_2| << |\lambda_1| \rightarrow$ converge quickly.

▶ Shifted Inverse power iter.

converge to λ_{min} if shift $= 0$.

Special case : $\lambda_1 = \lambda_2$.

$\downarrow$

converge to $\lambda_1 (\alpha_1 \vec{v}_1 + \alpha_2 \vec{v}_2)$.

→ Spectral Radius.

$$\rho(A) = |\lambda_{max}|.$$

$$\|A\| \geq \rho(A)$$

- Assume $\rho(A) = |\lambda_k|$.
- define $V = [\vec{v}_k \quad \vec{v}_k \quad \cdots \quad \vec{v}_k]$ $\mathbb{R}^{n \times n}$
- $\|AV\| \leq \|A\| \|V\|$.
- $AV = [A\vec{v}_k \quad A\vec{v}_k \quad \cdots \quad A\vec{v}_k] = \lambda_k V$.

and $\|\lambda_k V\| = \|\lambda_k\| \|V\|$ $= \rho(A) \|V\|$.

- therefore $\|AV\| = \rho(A) \|V\|$

$$\leq \|A\| \|V\|.$$

$$\rightarrow \rho(A) \leq \|A\|$$

Spectral radius is a bound.

symmetric, diagonizable $A \in \mathbb{R}^{n \times n}$.

$$A = V \Lambda V^{-1}.$$

$$\hookrightarrow \begin{cases} \det(AB) = \det(A)\det(B). \\ \det(A^{-1}) = 1 / \det(A). \\ A \text{ or } B \leftarrow \text{diagonal / triangular} \end{cases}$$

$$\det(A) = \det(V \Lambda V^{-1})$$

$$= \det(V) \det(\Lambda) \det(V^{-1})$$

$$= \det(\Lambda). = \prod_i \lambda_i$$

If any $\lambda_i = 0$ → matrix is not invertible !

☆☆☆

Date.

No.

Diagonalizability.

diag: $A = V \Lambda V^{-1}$ exists

no connection

invert.: A^{-1} exists.

examples. diag. & invert.

Invert. → relates to eigenvals.

diag. → eigenvectors

Spectral Representation of a Matrix.

preview of SVD.

Symmetric & diagonizable. $A \in R^{n \times n}$.

$$A = V \Lambda V^{-1} = V \Lambda V^{T}.$$

$$\rightarrow A = \sum_{i=1}^{n} \lambda_i \vec{v}_i \vec{v}_i^{T}.$$

Remark: $\vec{v}_i \cdot \vec{v}_i^{T} \neq \vec{v}_i^{T} \vec{v}_i$.

Date.

No.

$\vec{v}_i^{T} \vec{v}_i$ ⇒ inner product.

$\vec{v}_i \vec{v}_i^{T}$ ⇒ outer product.

Week 8-2

$n \times n$ matrix A.

$Q, R = qr(A)$.

$A \leftarrow QR$.

Mathematically,

$A \rightarrow$ real non-singular matrix.

$A, B \rightarrow$ similar matrices.

exist non-singular matrix T

$\downarrow$

$$A = TBT^{-1}$$

if A & B are similar matrices,

They have the same eigenvalues.

$$\det(A - \lambda I) = \det(TBT^{-1} - \lambda I).$$

$$= \det(T)\det(B - \lambda I)\det(T^{-1})$$

$$= \det(B - \lambda I).$$

$$TAT^{-1}y = \lambda_2 y$$

$$A^{(1)} = \varphi^T A \varphi$$

$$A^{(1)} = \varphi^{(1)} R$$

$$A^{(2)} = R^{(1)} \varphi^{(1)}.$$

$\downarrow$

$$A^{(k)} = \underbrace{\varphi^{(k-1)} \varphi^{(k-2)} \cdots}_{(\hat{Q})^T} A$$

$$(\hat{Q})^T = (\hat{\varphi})^{-1} \quad \underbrace{\varphi^{(1)} \cdots \varphi^{(k-1)}}_{\tilde{\varphi}}.$$

$$A^{(k)} = \hat{Q} A Q$$

Date.

No.

$$A^m = Q^m R^n$$

$$A^{m+1} = R^{(n+1)} Q^{n+1}$$

QR.

$$A^{(1)} = Q^T A Q = Q^{-1} A Q$$

LHS: RHS.

$$Q^{(1)} R^{(1)} = A^{(1)}.$$

$$A^{(2)} = R^{(1)} Q^{(1)}$$

$$\vdots$$

$$A^{(k)} = Q^{k-1} Q^{k} \cdots Q^{(1)} A Q^{(1)} \cdots Q^{k-1} Q$$

$$A^{(2)} = \left(Q^{(1)}\right)^{-1} A^{(1)} Q^{(1)}$$

$$\left(Q^{(1)}\right)^{-1} \left(Q^{-1} A Q\right) Q^{(1)}.$$

Date.

No.

Cayley-Hamilton Theorem

$$A \rightarrow \det(A - \lambda I).$$

$$= A^m + \alpha_{m-1} \lambda^{m-1}$$

$$+ \alpha_{m-2} \lambda^{m-2} + \cdots \alpha_1 A + \alpha_0 = 0$$

$$P A = A^m + \alpha_{m-1} A^{m-1} + \cdots + \alpha_1 A$$

$$+ \alpha_0 I = 0$$

A diagonalizabb → $A = T \Lambda T^{-1}.$

↳ diagonal matrix.

$$P(\Lambda) = \Lambda^m + \alpha_{m-1} \Lambda^{m-1} + \cdots$$

$$+ \alpha_0 I = 0$$

each diagonal elem. eigenvals of A.

Date.

No.

$A \rightarrow$ diagonalizable

$$A = \overset{T}{\begin{bmatrix} | \\ x_i \\ | \end{bmatrix}} \overset{\Lambda}{\begin{bmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \ddots \\ & & & \lambda_n \end{bmatrix}} \overset{T^{-1}}{\begin{bmatrix} - & x_i & - \end{bmatrix}}$$

$$A = \underbrace{T \Lambda T^{-1} = Q S Q^{-1}}_{\text{find eigenvals \& eigenvec.}}$$

$$A^{(3)} = \alpha .$$

$$A^{(3)} = \alpha_2 A^2 + \alpha_1 A^1 + \alpha_0 I .$$

$$\sqrt{A} = P(\sqrt{A})^2 + PA = P_0 I$$

$$\sqrt{\lambda_1} = P_0 [\ ]^2 + R_1 \lambda + P_0$$

$$\sqrt{\lambda_2} = P_1 \quad \cdots$$

$$\sqrt{\lambda_3} = P_2 \quad \cdots$$

Week 8 - 3.

Date.

No.

$$A = \begin{bmatrix} 2 & 1 & -4 \\ 0 & 1 & 8 \\ 0 & 0 & 4 \end{bmatrix}$$

ⓐ find the characteristic polynomial

ⓑ Find a form for A^{-1} in terms of powers of A

ⓒ Calculate A^{-1} and characterize

$$AA^{-1} = I$$

ⓐ

$$(2-\lambda)(1-\lambda)(4-\lambda) = 0$$

$$-A^3 - 7A^2 - 14A + 8I = 0.$$

$$A^{-1} = \frac{1}{8}\left[A^2 - 7A + 14\right].$$

Date.

No.

$$A^{-1} = \begin{bmatrix} 1/2 & -1/2 & 3/2 \\ 0 & 1 & -2 \\ 0 & 0 & 1/4 \end{bmatrix}$$

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

① Find values of $c \Rightarrow A$ postive definite

② For $c=0$

$\downarrow$ eigenvalues, eigenvectors decouple

Find A^{10} using coupling

Find A^{10}, using Caley-Hamilton

① postive definite →

$$= (c-\lambda)^3 - 4(c-\lambda) - 4(c-\lambda)$$

$$= (c-\lambda)\left[(c-\lambda)^2 - 8\right]$$

decoupling $A = X \Lambda X$ Date.

No.

$\lambda = c, \quad c = \sqrt{8}$

$\downarrow$

$c > \sqrt{8}$

$\lambda_1 = \sqrt{8}, \quad \lambda_2 = 0, \quad \lambda_3 = -\sqrt{8}$

$Av_2 = \lambda v_2 \quad \ldots \quad Av_1 = \lambda v_1$

$$\begin{bmatrix} 0 & 2 & 0 \\ 2 & 0 & 2 \\ 0 & 2 & 0 \end{bmatrix} \qquad \begin{bmatrix} -\sqrt{2} & 2 & 0 \\ 2 & -\sqrt{2} & 2 \\ 0 & 2 & \sqrt{2} \end{bmatrix}$$

$\downarrow \qquad \downarrow$

$$\vec{v} = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} \qquad \vec{v} = \begin{bmatrix} 1 \\ 8 \\ 1 \end{bmatrix}$$

$Ax_3 = \lambda v_3.$

$$\begin{bmatrix} \sqrt{8} & 2 & 0 \\ 2 & \sqrt{8} & 2 \\ 0 & 0r & \sqrt{8} \end{bmatrix} \rightarrow \vec{v} = \begin{bmatrix} 1 \\ -\frac{1}{\sqrt{2}} \\ 1 \end{bmatrix}$$

Date.

No.

$$A^n = \Phi \Lambda \Phi^{-1} \Phi \Lambda \Phi^{-1} \Phi \Lambda \Phi \cdots \Phi \Lambda \Phi^{-1}$$

$$\to \Phi \Lambda^n \Phi^{-1} \to e^A \to \Phi e^\Lambda \Phi^{-1}$$

$$A^{10} = \Bigg[$$

Final Week.

$\chi_A(z) = \det(A - zI)$.

QR iteration:

$$\det(B - zI) = \det(SAS^{-1} - zI)$$
$$= \det(S)\det(A - zI)\det(S^{-1})$$
$$= \det(A - zI)$$

Cayley-Hamilton Theorem.

Thm: (CH). Every matrix satisfies

Date.

No.

its characteristic polynomial.

i.e., if A is $n \times n$, then

$$\chi_A(A) = 0_{n\times n}.$$

w/ $\chi_A(z) = \det(A - zI)$ denoting the characteristic poly. of A.

Here, we want to write

$$\chi_A(z) = (z-\lambda_1)(z-\lambda_2)\cdots(z-\lambda_n)$$
$$= z^n + c_1 z^{n-1} + \cdots + c_n z + \det(A))$$

Proof: (sketch).

Suppose A is diagonalizable.

So $A = T \Lambda T^{-1}$ for an invertible T and diagonal Λ.

WTS: $\chi_A(A)\vec{x} = \vec{0}$ for all $\vec{x} \in \mathbb{R}^n$.

Keypoint: A is diagonalizable,

So its eigenvecs $\vec{v}_1, \vec{v}_2, \dots \vec{v}_n$

are a basis of $\mathbb{R}^n$.

So for any $\vec{x}$, we can write

$$\vec{x} = c_1\vec{v}_1 + \dots c_n\vec{v}_n$$

for some coeff c_j:

Since $\vec{v}_j$ is a λ_j-eigenvec,

$$A\vec{v}_j = \lambda_j \vec{v}_j$$

$$\to (A-\lambda_j)\vec{v}_j = 0$$

So now, we try on the char

poly.: $\chi_A(A) = (A-\lambda_1 I)(A-\lambda_2 I)\dots(A-\lambda_n I)$

$\downarrow$ evaluate this on $\vec{x}$.

Notice $\left[\prod_{j=1}^{n} (A-\lambda_j I)\right]\vec{x} = c_1\left[\prod_{j=1}^{n} (A-\lambda_j I)\right]v_1$

$+\dots + c_n\left[\prod_{j=1}^{n} (A-\lambda_j I)\right]v_n$

$= c_1(A-\lambda_2 I)\dots(A-\lambda_n I)(A-\lambda_1 I)v_1$

$+\dots + c_n(A_1-\lambda_1 I)\cdots(A-\lambda_n I)v_n$

Non-diagonalizable A: the argument:

(see generalized eigenvectors, Jordan b.)

Main reason: C-H still works is

b/c char. poly. has enough repeated

factors ...

Consider: $J = \begin{bmatrix}2 & 1\\ & 2\end{bmatrix}$ diagonalizability

eigenvals: 2, 2 no. of eigenvalue.

eigenvec: 1

example check.

$\chi_J(z) = (z-2)(z-2)$

$= (J-2I)(J-2I)\begin{bmatrix}0\\1\end{bmatrix}$

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

Date.

No.

Applications.

1) Inversion formula.

Example C-H implies $\chi_J(J)=(J-2I)^2 = 0$

So $J^2 - 4J + \boxed{4}I = 0.$

Rearrange to obtain inverse!

$$\Leftrightarrow J(J-4I) = -4I$$

multiply by J^{-1} to obtain a formula!

$$\Rightarrow (J-4I) = -4J^{-1}$$

$$\Rightarrow J^{-1} = \frac{1}{4}(J-4I).$$

Keypoint constant term is $\det(A)$

$$\det(A - zI)\Big|_{z=0} = \chi_A(0).$$

$$= c_1 z^{n-1} + \cdots + c_{n-2} z + c_{n-1}\Big|_{z=0}$$

2) Analytic fns of matrices. * (MG)

$$e^A = I + \cdots$$

$$= k_1 I + k_2 A + \cdots + k_n A^{n-1}$$

Krylov subspace methods

Additional Linear Algebra Notes

Determinants.

<u>Def'n</u>: Scalar function of matrix entries that can determine whether matrix is singular.

How to determine?

① $|x| \to \det(A) = A$.

② $\det\left(\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}\right) = (-1)\, a_{11} \det(a_{22})$
$+ (-1)^{1+2}\, a_{12} \det(a_{21})$
$= a_{11}a_{22} - a_{12}a_{21}$.

③ $\det\begin{pmatrix} a_{11} & a_{12} \cdots & a_{1n} \\ a_{21} & & \\ \vdots & & \\ a_{n1} & & a_{nn} \end{pmatrix} = \sum_{ij} (-1)^{i+j}\, a_{ij}\, |C_{ij}|$

(for $1 \le i \le n$)

C_{ij} is A w/ i^{th} row & j^{th} col'n eliminated

$\det\begin{vmatrix} 2 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{vmatrix} = 0.$

number of row flips

$\det(A) = 0$ iff A is singular $\to \det(A) = (-1)^k$

Update: $\begin{cases} k_i \to i+1. & \\ k_{i+1} \to i & k_{i+1} \to i+2 \\ k_{i+2} \to i+1 & k_{i+2} \to i+2. \\ \vdots & \\ k_{j-1} \to j & k_{j-1} \to j-2 \\ k_j \to j-1. & \end{cases}$

$b \in$ Col'n (A) $\iff$ $Ax=b$ has a solution.

$\dim(\mathcal{N}(A)) = n - rk(A)$

↳ tells num. DoF in systems sol'n

$\Rightarrow Ax=0.$ ↪ $0 \cdot x = 0.$

$\sum_i x_i \vec{u}_i = 0$ Same relation Col'n (A) ↔ col'n (0)

$\Rightarrow \dim(\text{col}(A)) + \dim(\mathcal{N}(A)) = n$

Every vector $\in \mathbb{R}^k$ is sum of the components (vector) in col(A) & $\mathcal{N}(A)$

basis of $\mathcal{N}(U) = \mathcal{N}(A)$.

▶ If vectors $\vec{x}_1, \dots, \vec{x}_m$ in $\mathbb{R}^n$ span subspace S in $\mathbb{R}^n$ $\dim(S)=m$.

False – it's not a basis. → redundant.

▶ If A, B have same row space, col'n space, Same null space → $A=B$.

False. counter-example: $A = \alpha B$.

▶ If $m \times n$ matrix A, $A\vec{x}=\vec{b}$ always has at least one sol'n for every choice of $\vec{b}$, then the only sol'n to $A^T\vec{y}=\vec{0}$ is $\vec{y}=\vec{0}$.

True.

↳ to have at least one sol'n for any $\vec{b}$, col'n space must be all of $\mathbb{R}^m$.

↓

$\vec{b}$ must be in $\mathbb{R}^m$

$r(A) = m.$

$\dim(\mathcal{N}(A)) = m - m = 0$, $r(A^T) = r(A).$

only sol'n: $\vec{y} = \vec{0}$, zero vector.

A $m \times n$ matrix, rank $r \leq \min\{m, n\}$.

(a). $A\vec{x} = \vec{b}$ has no sol'n regardless $\vec{b}$

Impossible for $\vec{b} = 0$. always $\vec{x} = 0$.

(b). $A\vec{x} = \vec{b}$ has exactly one sol'n for any $\vec{b}$.

IFF col'n of A form a basis space $\mathbb{R}^m$ (independent).

$m = n = r$ (A non-singular, square)

(c) $A\vec{x} = \vec{b}$ has infinitely many sol'n for any $\vec{b}$.

$\dim(N(A)) \geqslant 1$. → subspace containing only zero $\vec{0}$, dim is $\vec{0}$

$n - r \geqslant 1$.

$r = m$,

$m = r < n$.

null space .

(d). If $\vec{x}_i, i = 1, \dots, m$ are orthogonal. they are independent, $\vec{x}_i \in \mathbb{R}^n$, $n > m$.

True. $c_1\vec{x}_1 + \cdots + c_m\vec{x}_m = 0$

$$(c_1\vec{x}_1 + \cdots + c_m\vec{x}_m)\begin{pmatrix}\vec{x}_1 \\ \vec{x}_2 \\ \vdots \\ \vec{x}_m\end{pmatrix} = 0.$$

$$c_1\vec{x}_1^T\vec{x}_i + c_2\vec{x}_2^T\vec{x}_i + \cdots + c_m\vec{x}_m^T x_i = 0$$

All orthogonal :

$c_i\vec{x}_i^T\vec{x}_i = 0$ remains:

as $\vec{x}_i^T\vec{x}_i \neq 0$.

$c_i = 0$, for $i = 1, \dots, m$.

Q: Prove that $\|Ax\|_2 \leq \|A\|_2 \|x\|_2$.

1) $\|x\|_2 = \sqrt{\sum_{i=1}^{n} |x_i|^2}$

Induced norm:

2). $\|A\|_2 = \max\limits_{x \neq 0} \dfrac{\|Ax\|_2}{\|x\|_2}$ ... maximum over all possibility.

→ this is an "induced" norm, which means it measures how big the size of the output Ax can be, write the size of the input x.

If want to show,

for any vector $\vec{x}$, $\|Ax\|_2 \le \|A\|_2 \|x\|_2$.

(If $x \neq 0$), rewrite as

$$\frac{\|Ax\|_2}{\|x\|_2} \le \|A\|_2$$

If for any $x \neq 0$, we know that

$$\frac{\|Ax\|_2}{\|x\|_2} \le \max_{x \neq 0} \frac{\|Ax\|_2}{\|x\|_2} = \|A\|_2 .$$

So indeed,

$$\|Ax\|_2 \le \|A\|_2 \|x\|_2$$

Ill conditioning arises when matrix is close to singular conditioning.

⇒ Intuition for condition number.

Roughly, a matrix is ill-conditioned if it is "close" to being singular.

<u>Example</u>: Consider $A\vec{x} = \vec{b}$.

w/ $A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 2+\varepsilon & 2 \end{bmatrix}$ and

$b = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$, $\varepsilon \approx 0$, but $\varepsilon \neq 0$.

$$A\vec{x} = \vec{b} \leftrightarrow \begin{bmatrix} 1 & 1 \\ 2+\varepsilon & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} .$$

$$\hookrightarrow \begin{cases} x_1 + x_2 = 1 \\ (2+\varepsilon)\, x_1 + 2x_2 = 2 . \end{cases}$$

...

Notes on Determinants.

- Scalar function of matrices entries that can determine whether matrix is singular.

Inductive def'n:

① 1×1 mat A.

$$\det([a]) = a.$$

② $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$.

$$\det(A) = a \det([d]) - b \cdot \det([c]).$$

$$= ad - bc.$$

③ If $A = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & k \end{bmatrix}$.

$$\det(A) = a \cdot \det\left(\begin{bmatrix} e & f \\ h & k \end{bmatrix}\right) - b \cdot \det\left(\begin{bmatrix} d & f \\ g & k \end{bmatrix}\right)$$

$$+ c \cdot \det\left(\begin{bmatrix} d & e \\ g & h \end{bmatrix}\right).$$

$$= a(ek - hf) + (-b)(dk - gf) + c(dh - ge)$$

but.

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

$\det(A) = 0$, A is not invertible.

Def'n. The determinant of $m \times n$ A:

$$\det(A) = \sum_{j=1}^{n} (-1)^{j+i} a_{ij} \det(M_{ij}).$$

where M_{ij} is the sub-mat obtained from A by removing its i^{th} row & j^{th} col'n.

Note: Can use any row we like!

Example: Using A as in previous example: expand along second row.

$$\det(A) = 0 = -0 \cdot \det(M_{21}) + 1 \cdot \det(M_{22})$$

$$+ 0 \cdot \det(M_{23}) \times (-1).$$

$$= -\det\left(\begin{bmatrix} 2 & 0 \\ 1 & 0 \end{bmatrix}\right) = 0$$

* Notation vertical bars mean det

i.e., $\det\left(\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}\right) = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}$.

Properties

1) Scaling.

$$\begin{vmatrix} t\cdot a & b \\ t\cdot c & d \end{vmatrix} = t \begin{vmatrix} a & b \\ c & d \end{vmatrix}.$$

2). Additive

$$\begin{vmatrix} a+v & b \\ c+w & d \end{vmatrix} = \begin{vmatrix} a & b \\ c & d \end{vmatrix} + \begin{vmatrix} v & d \\ w & d \end{vmatrix}.$$

Note: Second col'n the same!

3) Alternating.

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix} = - \begin{vmatrix} b & a \\ d & c \end{vmatrix}.$$

Swapping col'ns negats determinant.

Q: what happens if two col'ns are scalar multiple of each other?.

$$\begin{vmatrix} ta & a \\ tc & c \end{vmatrix} = t \begin{vmatrix} a & a \\ c & c \end{vmatrix}.$$

A. That matrix has determinant zero!

$$t\begin{vmatrix} a & a \\ c & c \end{vmatrix} = -t\begin{vmatrix} a & a \\ c & c \end{vmatrix} = 0$$

Consequence: Using additivity, the last result can be extended to general linear combination.

Theorem: det(A) = 0, IFF A is singular.

Roughly, det(A) "detects" linear combination in columns of A.

Fact: ① $|A| = |A^T|$ for any nxn A matrix.

Consequence: we have all the same prop. for rows.

(e.g. swaps, scalars, multi, addi.)

Keypoint: We can compute det(A) by performing Gauss elimination and keeping track of the row operation.

det(U) is easy to compute.

Example: $\begin{vmatrix} a & b \\ c-\alpha a & d-\alpha b \end{vmatrix}$.

$$= \begin{vmatrix} a & b \\ c & d \end{vmatrix} + \cancel{\begin{vmatrix} a & b \\ -\alpha a & -\alpha b \end{vmatrix}} \rightarrow \text{Same row OR col'n}$$

$$= \begin{vmatrix} a & b \\ c & d \end{vmatrix}$$

(7) $|AB| = |A||B|$.

The determinant of a product is the product of determinants.

Example: Show that if P is a projection matrix.

Example: Show that $\det(I) = 1$.

Proof: let A be any non-singular $n \times n$ matrix.

$\because A \cdot I = A$

$\therefore |AI| = |A||I| = |A|$.

$\because |\det(A)| \neq 0$, $|I| = 1$

Recall

Theorem: Given any $m \times n$ matrix A;

lin. indp. rows = lin. indp. col'ns.

$$A = \begin{bmatrix} \quad \end{bmatrix} \in \mathbb{R}^{\underset{\substack{\downarrow \\ \text{row}}}{1000} \times \underset{\substack{\downarrow \\ \text{col'n}}}{100}}$$

rows in $\mathbb{R}^{100}$; col'ns in $\mathbb{R}^{1000}$

In symbols,

dim row(A) = dim col'n(A).

Example

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 5 & 4 \\ 3 & 7 & 7 \end{bmatrix}.$$

Using Gaussian Elimination, find $A = LU$.

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

Keypoints : GE preserves A row space

$$\text{row}(A) = \text{row}(U)$$

Keypoints Read off basis of row(A) from U.

As GE destroys col'ns we can't tell. But using th'm, we know how big that col'n space has to be.

i.e., $\dim(\text{col}(A)) = \text{rank}(A)$.

Here, 2 linearly independent rows in U.

$\dim(\text{row}(U)) = \dim(\text{row}(A)) = 2$
$= \dim(\text{col'n}(A))$.

Since $\text{rank}(A) = 2$, and first 2 col'ns are linearly indp.

$$\text{col}(A) = \text{span}(\vec{a}_1, \vec{a}_2).$$

In general,

$$A\vec{x} = 0 \leftrightarrow U\vec{x} = 0$$

$$x_1 U_1 + x_2 U_2 + \dots + x_n U_n = 0.$$

So, same linear relation exist between the col'ns of U & col'ns of A.

→ pick out col'ns of A corresponding to input col'ns of U.

Q. What is $N(A)$

e.g., set of x, s.t. $A\vec{x} = 0$.

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 5 & 4 \\ 3 & 7 & 7 \end{bmatrix} \vec{x} = 0$$

Recall:

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

$$\begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & -2 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

2nd row : $x_2 = 2x_3$.

1st row : $x_1 = -2x_2 - 3x_3$
$= -7x_3$.

Key observation: 2 eqns. $\rightarrow \begin{cases} \text{rank}(A) = 2. \\ \dim(\text{col'n}(A)) = 2 \end{cases}$

1 degree of freedom for Null space.

Theorem

$$\dim(\text{col'n}(A)) + \dim(N(A)) = n.$$

for any $m \times n$ matrix A.

thus, every vector in $\mathbb{R}^n$ is the same of the two components.

i.e., a vector in col(A)

& a Vector in $N(A)$

→ Consequence $A\vec{x} = \vec{b}$,

- If $b \in$ col(A) ↔ $A\vec{x} = \vec{b}$ has a sol'n.

- $\text{Dim}(N(A)) = n - \text{rank}(A)$.

tells you # of DoFs in sol'n to $A\vec{x} = \vec{b}$.

ME300A HW#1
Hanfeng Zhai (hzhai@stanford.edu) October 7, 2023

Problem 1. *Decide whether each of the following statements is true or false. If true, then prove it; otherwise, provide a counterexample.*

(a) If $AB = I$, then $A = I$.

Solution. Counterexample: $B = A^{-1}$. □

(b) If $AB = 0$, then A or B is a zero matrix.

Solution. Counterexample:

$$A = \begin{bmatrix} a & a \\ a & a \end{bmatrix}, B = \begin{bmatrix} b & -b \\ -b & b \end{bmatrix} \tag{1}$$

where a and b are non-zero scalars. □

(c) If AB and BA are defined, then both A and B must be square.

Solution. Counterexample: A is a 2×3 matrix, and B is a 3×2 matrix. More generally, A is a $m \times n$ matrix, and B is a $n \times m$ matrix. □

(d) If AB and BA are defined, then both AB and BA are necessarily square.

Solution. Assume A is a $m \times n$ matrix, and B is a $n \times m$ matrix. Since both AB and BA are defined, assume $AB = \mathcal{A}$, and $BA = \mathcal{B}$, then $\mathcal{A}$ has dimension $n \times n$, and $\mathcal{B}$ has dimension $m \times m$. Assume A and B can be generalized to two second-order tensors, using indicial notation:

$$\begin{aligned} A_{ij}B_{ji} &= \mathcal{A}_{ii}, \quad i \in [1, m], j \in [i, n] \\ B_{ji}A_{ij} &= \mathcal{B}_{jj}, \quad i \in [1, m], j \in [i, n] \end{aligned} \tag{2}$$

Hence, both AB and BA are necessarily square. □

(e) If A is invertible, then $(A^{-1})^T = (A^T)^{-1}$.

Solution. If A is invertible, then $\left(A^{-1}\right)^\mathsf{T} = \left(A^\mathsf{T}\right)^{-1}$. We further get $\left(A^{-1}\right)^\mathsf{T} A^\mathsf{T} = I$, then we finally get:

$$\left(A^{-1}A\right)^\mathsf{T} = I \tag{3}$$

then the relation $(A^{-1})^T = (A^T)^{-1}$ is established. □

Problem 2. *Suppose A and B are $n \times n$ symmetric matrices; that is, $A = A^T$ and $B = B^T$. Decide whether each of the following matrices is symmetric. If it is, prove it; otherwise, provide a counterexample.*

(a) $A^2 - B^2$.
Solution.

$$\begin{aligned}(A^2 - B^2)^\intercal &= ((AA) - (BB))^\intercal \\ &= (AA)^\intercal - (BB)^\intercal \\ &= A^\intercal A^\intercal - B^\intercal B^\intercal \\ &= (AA) - (BB) \\ &= A^2 - B^2\end{aligned} \tag{4}$$

□

(b) $(A + B)(A - B)$.
Solution.

$$\begin{aligned}[(A + B)(A - B)]^\intercal &= (A - B)^\intercal (A + B)^\intercal \\ &= (A^\intercal - B^\intercal)(A^\intercal + B^\intercal) \\ &= (A - B)(A + B)\end{aligned} \tag{5}$$

A counterexample would be $A = \begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix}$, $B = \begin{bmatrix} 7 & 5 \\ 5 & 6 \end{bmatrix}$, then $(A - B)(A + B) = \begin{bmatrix} -69 & -66 \\ -54 & -51 \end{bmatrix}$, and $(A + B)(A - B) = \begin{bmatrix} -69 & -54 \\ -66 & -51 \end{bmatrix}$, where $(A - B)(A + B) \neq (A + B)(A - B)$. □

(c) $ABAB$.
Solution.

$$\begin{aligned}[ABAB]^\intercal &= (AB)^\intercal (AB)^\intercal \\ &= B^\intercal A^\intercal B^\intercal A^\intercal \\ &= BABA\end{aligned} \tag{6}$$

Using the same counterexample from (b), we get $ABAB = \begin{bmatrix} 713 & 672 \\ 924 & 881 \end{bmatrix}$, and $BABA = \begin{bmatrix} 713 & 924 \\ 672 & 881 \end{bmatrix}$. It is found that $ABAB \neq BABA$, hence the statement is wrong. □

(d) ABA.
Solution.

$$\begin{aligned}[ABA]^\intercal &= (A)^\intercal (AB)^\intercal \\ &= A^\intercal B^\intercal A^\intercal \\ &= ABA\end{aligned} \tag{7}$$

The statement is true. □

Problem 3. *A square matrix A is called right stochastic if the elements in each row have a unit sum. That is, a given $n \times n$ matrix A is right stochastic if*

$$\sum_{j=1}^{n} a_{ij} = 1,$$

, for each $1 \leq i \leq n$. Suppose A and B are $n \times n$ right stochastic matrices. Show that AB is right stochastic.

Solution. Assuming both A and B are right stochastic, by expanding AB we get

$$AB = \begin{bmatrix} \underbrace{\overbrace{a_{11}b_{11} + a_{12}b_{12} + ... + a_{1n}b_{1n},}^{1^{\text{st}}\ \text{term}} \quad \overbrace{a_{11}b_{21} + ... + a_{1n}b_{2n},}^{2^{\text{nd}}\ \text{term}} \quad \overbrace{a_{11}b_{n1} + ... + a_{1n}b_{nn}}^{\text{n}^{\text{th}}\ \text{term}}}_{1^{\text{st}}\ \text{row}} \\ \underbrace{\overbrace{\sum_{j=1}^{n} a_{2j}b_{1j},}^{1^{\text{st}}\ \text{term}} \quad \overbrace{\sum_{j=1}^{n} a_{2j}b_{2j},}^{2^{\text{nd}}\ \text{term}} \quad \overbrace{\sum_{j=1}^{n} a_{2j}b_{nj}}^{\text{n}^{\text{th}}\ \text{term}}}_{2^{\text{nd}}\ \text{row}} \\ ... \\ \underbrace{\overbrace{\sum_{j=1}^{n} a_{ij}b_{1j},}^{1^{\text{st}}\ \text{term}} \quad \overbrace{\sum_{j=1}^{n} a_{ij}b_{2j},}^{2^{\text{nd}}\ \text{term}} \quad \overbrace{\sum_{j=1}^{n} a_{ij}b_{nj}}^{\text{n}^{\text{th}}\ \text{term}}}_{i^{\text{th}}\ \text{row}} \\ ... \\ \underbrace{\overbrace{\sum_{j=1}^{n} a_{nj}b_{1j},}^{1^{\text{st}}\ \text{term}} \quad \overbrace{\sum_{j=1}^{n} a_{nj}b_{2j},}^{2^{\text{nd}}\ \text{term}} \quad \overbrace{\sum_{j=1}^{n} a_{nj}b_{nj}}^{\text{n}^{\text{th}}\ \text{term}}}_{n^{\text{th}}\ \text{row}} \end{bmatrix} \tag{8}$$

Since both A and B are right stochastic, we know $\sum_{j=1}^{n} a_{ij} = 1$ and $\sum_{j=1}^{n} b_{ij} = 1$, therefore $\sum_{j=1}^{n} \sum_{j=1}^{n} a_{ij}b_{ij} = \underbrace{(a_{i1} + a_{i2} + ... + a_{ij})}_{\equiv 1} \underbrace{(b_{i1} + b_{i2} + ... + b_{ij})}_{\equiv 1} = 1$. Hence AB is right stochastic. □

Problem 4. *Consider the system of equations $Ax = b$, with*

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 0 & 2 & 3 \\ 3 & 4 & 4 & 5 \\ 4+\epsilon & 5 & 4 & 5 \end{bmatrix}, \quad \text{and } b = \begin{bmatrix} 10 \\ 17 \\ 43 \\ 46+\epsilon \end{bmatrix}$$

(a) Show that if $\epsilon \neq 0$, the correct solution is $x1 = 1$, $x2 = 2$, $x3 = 3$, and $x4 = 4$. In addition, show that if $\epsilon = 0$, the vector $x^* = [1\ 2\ 3\ 4]^\mathsf{T}$ is still a solution (but not the only one). Find a linear relationship between the rows of A in this case.

Solution. We can first solve for x by doing the inverse of A:

$$A^{-1} = \frac{1}{\epsilon} \begin{bmatrix} 1 & \frac{1}{2} & -\frac{3}{2} & 1 \\ -(\epsilon+1) & -\frac{\epsilon+1}{2} & \frac{\epsilon+3}{2} & -1 \\ 6\epsilon - 1 & \frac{\epsilon-1}{2} & \frac{-3(\epsilon-1)}{2} & -1 \\ -(4\epsilon-1) & \frac{1}{2} & \frac{2\epsilon-3}{2} & 1 \end{bmatrix} \tag{9}$$

we then get:

$$\begin{aligned} x &= A^{-1}b \\ &= \begin{bmatrix} \frac{1}{\epsilon}(\epsilon+46) - \frac{1}{\epsilon}46 \\ \frac{1}{2\epsilon}\left[43(\epsilon+3) - 37(\epsilon+1)\right] - \frac{1}{\epsilon}(\epsilon+46) \\ \frac{10}{\epsilon}(6\epsilon-1) - \frac{56}{\epsilon}(\epsilon-1) - \frac{1}{\epsilon}(\epsilon+46) \\ \frac{43}{2\epsilon}(2\epsilon-3) - \frac{10}{\epsilon}(4\epsilon-1) + \frac{17}{2\epsilon} + \frac{1}{\epsilon}(\epsilon+46) \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} \end{aligned} \tag{10}$$

If $\epsilon = 0$, substitute it back to Eq. (10) we can still get $x = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$. Hence, $x = x^*$ is still one of the solutions.

However, when $\epsilon = 0$ the equation to be solved becomes

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 0 & 2 & 3 \\ 3 & 4 & 4 & 5 \\ 4 & 5 & 4 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 10 \\ 17 \\ 43 \\ 46 \end{bmatrix} \tag{11}$$

in which the rank of the matrix A[1] is 3, indicating that the system is underdetermined, where the system possesses an infinite set of solutions.

We may further identify a linear relationship between the rows of A. Assuming the first three rows possess constants α, β, γ, and the linear combination of the first three rows is the fourth row. We can then obtain a new linear system to be solved:

$$\begin{bmatrix} 1 & -1 & 3 \\ 1 & 0 & 4 \\ 1 & 2 & 4 \\ 1 & 3 & 5 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = \begin{bmatrix} 4 \\ 5 \\ 4 \\ 5 \end{bmatrix} \tag{12}$$

[1] obtained using MATLAB `rank`

We can then solve to get $\alpha = -2$, $\beta = -1$, $\gamma = 3$. We can further contend that the linear combination takes the form

$$-2\begin{bmatrix}1\\1\\1\\1\end{bmatrix} - 1\begin{bmatrix}-1\\0\\2\\3\end{bmatrix} + 31\begin{bmatrix}3\\4\\4\\5\end{bmatrix} = \begin{bmatrix}4\\5\\4\\5\end{bmatrix} \tag{13}$$

□

(b) Use MATLAB to solve the system for $\epsilon = 10^{-k}$ and $k = 1, 2, ..., 15$. Plot the error in the numerical solution, given as the norm $\|x_{\text{numerical}} - x_{\text{exact}}\|$, and discuss the accuracy of your results.

Solution. To solve this problem, I wrote the following MATLAB codes:

```
err = [];
for k=1:1:15
    eps = 10^(k);
    A = [1,1,1,1;-1,0,2,3;3,4,4,5;4+eps,5,4,5]
    b = [10,17,43,46+eps]'
    x = A\b;
    x_bench = [1;2;3;4];
    err_x = norm(x-x_bench);
    err(k)=err_x;
end
```

By plotting the $\|x_{\text{numerical}} - x_{\text{exact}}\|$ (named "Norm") versus the k value, Fig. 1 is plotted on a log scale for the norm.

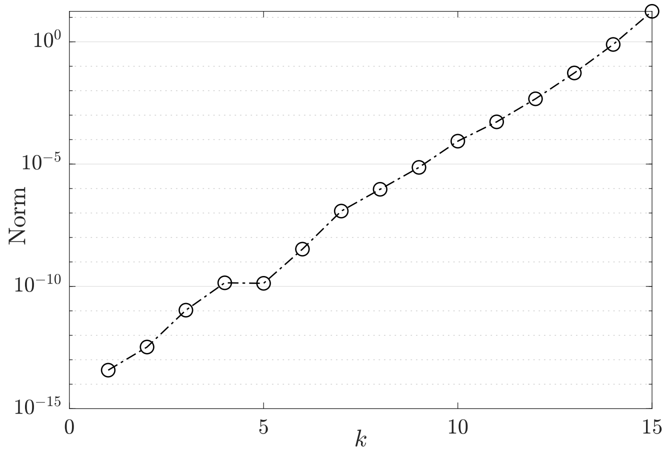


Figure 1: Norm-k curve for comparing the numerical and analytical solutions.

One deduces that with an increasing k value, the norm increases exponentially (realized by the "pseudo-linear" trend on the log scale). With an increasing k value, ϵ decreases in an exponential fashion, leading to the A matrix approximating the $\epsilon = 0$ scenario. We already know that when $\epsilon = 0$ matrix A is not fully ranked, leading to non-unique solutions. This explains when k increases, one observes an increasing error in the exponential fashion. □

Problem 5. *Consider 3 rectangular matrices*

$$A \in \mathbb{R}^{m\times n}, \quad B \in \mathbb{R}^{n\times k}, \quad C \in \mathbb{R}^{k\times l}$$

(a) What is the computational cost of computing $(AB)C$?
Solution. The computational burden is

$$m \times k \times (2n - 1) + m \times l \times (2k - 1) \tag{14}$$

The computational complexity of this operation is then either $\mathcal{O}(mnk)$ or $\mathcal{O}(mlk)$, which are both $\mathcal{O}(\mathrm{n}^3)$.□

(b) What is the computational cost of computing $A(BC)$?
Solution. The computational burden is

$$n \times l \times (2k - 1) + m \times l \times (2n - 1) \tag{15}$$

The computational complexity of this operation is then either $\mathcal{O}(mnk)$ or $\mathcal{O}(mlk)$, which are both $\mathcal{O}(\mathrm{n}^3)$.□

(c) Which method would you use to calculate the product of 3 matrices to minimize the computational cost?

Solution. One may realize the order of computational complexity for the two methods:

1. $\mathcal{O}\left((AB)C\right) = \mathcal{O}\left(mnk\right)$ or $\mathcal{O}\left(mlk\right)$.
2. $\mathcal{O}\left(A(BC)\right) = \mathcal{O}\left(nlk\right)$ or $\mathcal{O}\left(mln\right)$

Among the dimensions $m, n, l, \&\ k$, if the smallest value is

m : $\mathcal{O}\left(ABC\right)_{\text{min}} = \mathcal{O}\left(nlk\right)$, I would pick the method $A(BC)$.
n : $\mathcal{O}\left(ABC\right)_{\text{min}} = \mathcal{O}\left(mlk\right)$, I would pick the method $(AB)C$.
l : $\mathcal{O}\left(ABC\right)_{\text{min}} = \mathcal{O}\left(mnk\right)$, I would pick the method $(AB)C$.
k : $\mathcal{O}\left(ABC\right)_{\text{min}} = \mathcal{O}\left(mln\right)$, I would pick the method $A(BC)$.

□

Problem 6. *A closed economic model involves a society in which all the goods and services produced by members of the society are consumed by those members. No goods or services are imported from without and none are exported. Such a system involves N members, each of whom produces goods or services and charges for their use. The problem is to determine the prices each member should charge for their labor so that everyone breaks even after one year. For simplicity, we assume each member produces one unit per year.*

Consider a simple closed system limited to a farmer, a carpenter, and a weaver so that $N = 3$. Let p_1 denote the farmer's annual income (that is, the price she charges for her unit of food), let p_2 denote the carpenter's annual income, and let p_3 denote the weaver's. On an annual basis, the farmer and the carpenter consume 35% each of the available food, while the weaver consumes the remaining 30%. In addition, the carpenter uses 20% of the wood products he makes, while the farmer uses 35%, and the weaver uses the remaining 45%. The farmer uses 45% of the weaver's clothing, the carpenter uses 30%, and the weaver himself consumes the remaining 25%.

(a) Write down the break-even equations for the farmer, the carpenter, and the weaver.
Solution. We can first tabulate the consumption for farmer, carpenter, and weaver:

	food [%]	wood [%]	clothing [%]
farmer	35	35	45
carpenter	35	20	30
weaver	30	45	25

We can then write out the equation sets for money balance for farmer, carpenter, and weaver:

$$\begin{aligned} \text{farmer}: &\quad 0.65p_1 - 0.35p_2 - 0.45p_3 = 0 \\ \text{carpenter}: &\quad -0.35p_1 + 0.8p_2 - 0.3p_3 = 0 \\ \text{weaver}: &\quad -0.3p_1 - 0.45p_2 + 0.75p_3 = 0 \end{aligned} \tag{16}$$

□

(b) Express your system of break-even equations as a homogeneous matrix equation and solve it using MATLAB to find the break-even prices p_1, p_2, p_3.
Solution.

We can then solve for the linear equation $\begin{bmatrix} 0.65 & -0.35 & -0.45 \\ -0.35 & 0.8 & -0.3 \\ -0.3 & -0.45 & 0.75 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$. The solution for p is then $\begin{bmatrix} 0.6586 \\ 0.4993 \\ 0.5630 \end{bmatrix}$[2]. □

[2] found through using the MATLAB `null()` function.

ME300A HW#2
Hanfeng Zhai (hzhai@stanford.edu) October 21, 2023

Problem 1. *Determine which of the following sets are vector spaces. If you think a set is a vector space, prove it. If not, identify at least one vector space property that fails to hold.*

Recall that to prove a set is a vector space, it is sufficient to show it is a subspace of a known vector space.

Note In this problem, I will consider the vectors symbolized as u, v, w[1] in vector space V.

1. *The set of all 2×2 matrices $A = [a_{ij}]$ with $a_{11} = -a_{22}$ under standard matrix addition and scalar multiplication.*

   Solution. The set is a vector space. To prove it is a subspace of a known vector space, we recall the definition of a subspace:

   - The zero vector is contained in the set V.
   - $u + v \in V$.
   - $v \in \mathbb{R}$, $c \in \mathbb{R} \rightarrow cv \in \mathbb{R}$.

   Assuming there are two matrices in the defined set, A^I, $A^{II} \in V_A$. One may test the definitions respectively.

   - The zero matrix $A_0 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \in V_A$. It can be deduced that the first definition holds.
   - $A^\star = A^I + A^{II} = \begin{bmatrix} a^I_{11} & a^I_{12} \\ a^I_{21} & -a^I_{11} \end{bmatrix} + \begin{bmatrix} a^{II}_{11} & a^{II}_{12} \\ a^{II}_{21} & -a^{II}_{11} \end{bmatrix} = \begin{bmatrix} a^I_{11} + a^{II}_{11} & a^I_{11} + a^{II}_{12} \\ a^I_{11} + a^{II}_{21} & -a^I_{11} - a^{II}_{11} \end{bmatrix}$. Note that for $A^\star$ the defined property of the set also holds, i.e., $a^\star_{11} = -a^\star_{22}$. Hence, the second definition holds.
   - $A^\dagger = cA^I = \begin{bmatrix} ca^I_{11} & ca^I_{12} \\ ca^I_{21} & -ca^I_{11} \end{bmatrix}$. For the matrix $A^\dagger$, the vector set property preserves, i.e. $a^\dagger_{11} = -a^\dagger_{22}$. Hence, the third definition holds.

   Since the three definitions of a subspace to a known vector space hold, it is hence proven that the A is a vector space.□

2. *The set of all 3×3 upper triangular matrices under standard matrix addition and scalar multiplication.*

   Solution. This set is a vector space. Recall the definitions of a subspace to a known vector space from #1. We can first represent the set as M, where $M = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ 0 & m_{22} & m_{23} \\ 0 & 0 & m_{33} \end{bmatrix}$. We hence test the definitions of the vector space based on the subspace definition:

[1] stand for the more precise presentation as $\vec{u}$, $\vec{v}$, $\vec{w}$

- The zero matrix $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ agrees with the definition. Hence the first definition holds.
- The matrix addition $M^\star = M^I + M^{II} = \begin{bmatrix} m^I_{11} & m^I_{12} & m^I_{13} \\ 0 & m^I_{22} & m^I_{23} \\ 0 & 0 & m^I_{33} \end{bmatrix} + \begin{bmatrix} m^{II}_{11} & m^{II}_{12} & m^{II}_{13} \\ 0 & m^{II}_{22} & m^{II}_{23} \\ 0 & 0 & m^{II}_{33} \end{bmatrix}$ $= \begin{bmatrix} m^I_{11} + m^{II}_{11} & m^I_{12} + m^{II}_{12} & m^I_{13} + m^{II}_{13} \\ 0 & m^I_{22} + m^{II}_{22} & m^I_{23} + m^{II}_{23} \\ 0 & 0 & m^I_{33} + m^{II}_{33} \end{bmatrix}$ The new matrix $M^\star$ also agrees with the property of the upper triangular matrix. Hence definition 2 still holds.
- For scalar multiplication, $M^\dagger = cM = \begin{bmatrix} cm_{11} & cm_{12} & cm_{13} \\ 0 & cm_{22} & cm_{23} \\ 0 & 0 & cm_{33} \end{bmatrix}$. The new matrix still preserves the property of the upper triangular matrix, therefore the third definition of vector set still holds.

One can then conclude that the 3×3 upper triangular matrix preserves the properties of being a subspace to a known vector space. □

3. The set of all 3×3 lower triangular matrices of the form

$$\begin{bmatrix} 1 & 0 & 0 \\ a & 1 & 0 \\ b & c & 1 \end{bmatrix}$$

under standard matrix addition and scalar multiplication.

Solution. This is false. Considering the axiom $\mathcal{C}u \in V$[2]. If $\mathcal{C}$ is a non-one value, definition 1 is to be failed to hold:

$$\mathcal{C}u = \mathcal{C} \begin{bmatrix} 1 & 0 & 0 \\ a & 1 & 0 \\ b & c & 1 \end{bmatrix} = \begin{bmatrix} \mathcal{C} & 0 & 0 \\ a & \mathcal{C} & 0 \\ b & c & \mathcal{C} \end{bmatrix}$$

which violates the axiom of the original set. A simple counterexample could be when $\mathcal{C} = 5$:

$$\mathcal{C}u = \begin{bmatrix} 5 & 0 & 0 \\ a & 5 & 0 \\ b & c & 5 \end{bmatrix}$$

Hence, this is not a vector space, as it fails to hold to the property of $\mathcal{C}u \in V$, where V stands for the vector space. □

4. The set of all solutions to the linear system $Ax = b$, under standard vector addition and scalar multiplication.

[2] where $\mathcal{C}$ stands for a random constant.

Solution. This is false. Assuming the matrix A is invertible, one can represent the solution of the linear system as $V : x = A^{-1}b$ as the vector set. Now, consider the definition used in #3:

$$x' = \mathcal{C}x = \mathcal{C}A^{-1}b$$

According to the definition of a vector space, it should be obeyed that $x' \in V$. However, substituting x' one gets:

$$\begin{aligned} Ax' &= A\mathcal{C}A^{-1}b \\ &= \mathcal{C}AA^{-1}b \\ &= \mathcal{C}b \neq b, \ \text{(when } \mathcal{C} \neq 1) \end{aligned}$$

Hence, the axiom of $\mathcal{C}u \in V$ is violated, this is not a vector space. □

5. *The set of all degree* 2 *polynomials under standard polynomial addition and scalar multiplication.*

Solution. The set can be represented in the form $\{ax^2 + bx + c \mid x \in \mathbb{R}\}$. We consider the axiom of $u + v \in V$. Assuming there are two vector sets written as:

$$a_1x^2 + b_1x + c_1, \quad a_2x^2 + b_2x + c_2, \text{ with } x \in \mathbb{R}$$

If $a_1 = -a_2$, meanwhile $b_1 \neq -b_2$, the new system under addition will be

$$(b_1 - b_2)x + (c_1 - c_2)$$

which violates the definition of the degree 2 polynomial, i.e., $u + v \notin V$. What's more, if $a_1 = -a_2$ and $b_1 \neq -b_2$, the new system is

$$c_1 - c_2$$

which is just a constant, also does not agree with the degree 2 polynomial, i.e., $u+v \notin V$. Hence, this is not a vector space, from the previous two counterexamples.

□

Problem 2. 1. *Show that the matrix*

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

has no LU decomposition by writing out the equations corresponding to

$$A = \begin{bmatrix} l_{11} & 0 \\ l_{21} & l_{22} \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} \\ 0 & u_{22} \end{bmatrix},$$

and showing that the system has no solution.

Solution. One can first try to apply LU decomposition to the matrix A:

$$L = \begin{bmatrix} 1 & 0 \\ & 1 \end{bmatrix}, \quad U = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$$

By conducting row operations with multiplying factors, one tries to construct an updated U as an upper triangular matrix. Assuming the multiplying factor is λ:

$$L = \begin{bmatrix} 1 & 0 \\ \lambda & 1 \end{bmatrix}, \quad U = \begin{bmatrix} 0 & 1 \\ 1 & 1-\lambda \end{bmatrix}$$

It can be seen that $u_{21} = 1$ is independent of the value of λ, hence on cannot construct a upper triangular matrix for U, since the lower triangular part of U is a constant 1 independent of the row operation multiplier.

We can then proceed to further show the given system has no solution

$$\begin{aligned} A &= \begin{bmatrix} l_{11} & 0 \\ l_{21} & l_{22} \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} \\ 0 & u_{22} \end{bmatrix} \\ &= \begin{bmatrix} l_{11}u_{11} & l_{11}u_{12} \\ l_{21}u_{11} & l_{21}u_{12} + l_{22}u_{22} \end{bmatrix} \end{aligned}$$

To establish A, the relation $l_{11}u_{11} = 0$ has to be satisfied. Hence one obtains either $l_{11} = 0$ or $u_{11} = 0$.

If $l_{11} = 0$, then $l_{11}u_{12} = 0 \neq 1$, violating the original value in A. Hence, $l_{11} = 0$ is not a solution to this linear system.

If $u_{11} = 0$, then $l_{21}u_{11} = 0 \neq 1$, violating the original value in A. Hence, $u_{11} = 0$ is not a solution to this linear system.

Hence, $A = \begin{bmatrix} l_{11} & 0 \\ l_{21} & l_{22} \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} \\ 0 & u_{22} \end{bmatrix}$ has no solution. □

2. *Reverse the order of the rows of A and show that the resulting matrix does have an LU decomposition.*

Solution. After reversing the order, the new A is

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

which is an upper triangular matrix. One can then further apply the LU decompostion:

$$L = \begin{bmatrix} 1 & 0 \\ & 1 \end{bmatrix}, \quad U = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

Since U is already an upper triangular matrix, it is intuitive that $L = I$ establishes the LU relationship.

$$L = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad U = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

and hence the given statement is proved.

One may also prove this statement in the way provided in #1:

$$\begin{aligned} A &= \begin{bmatrix} l_{11} & 0 \\ l_{21} & l_{22} \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} \\ 0 & u_{22} \end{bmatrix} \\ &= \begin{bmatrix} l_{11}u_{11} & l_{11}u_{12} \\ l_{21}u_{11} & l_{21}u_{12} + l_{22}u_{22} \end{bmatrix} \equiv \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \end{aligned}$$

From $l_{21}u_{11} = 0$ we know that either $l_{21} = 0$ or $u_{11} = 0$. Since $l_{11}u_{11} = 1$, indicating that $u_{11} \neq 0$, therefore it has to be satisfied that $l_{21} = 0$.

Based upon this, we can further establish the relationship:

$$\begin{aligned} l_{11}u_{11} &= 1 \\ l_{11}u_{12} &= 1 \\ l_{22}u_{22} &= 1 \end{aligned}$$

It can be deduced that this system is solvable. One of the possible solutions is

$$l_{11} = l_{22} = u_{11} = u_{12} = u_{22} = 1$$

The statement is hence proved. □

Problem 3. *We say an $n \times n$ matrix A is strictly diagonally dominant (SDD) if*

$$|a_{i,i}| > \sum_{j \neq i} |a_{i,j}|$$

for each $i = 1, \ldots, n$.

Show that if A is SDD, it is also invertible.

Hint: Recall that A is invertible if and only if the linear system $Ax = 0$ has no non-trivial solutions.

Solution.

Based on the hint, we can first write out a N-dimensional linear system:

$$A\vec{x} = 0$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & & a_{2n} \\ & & & \vdots & \\ a_{n1} & a_{n2} & \ldots & & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = 0$$

$$\begin{bmatrix} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \\ \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n \end{bmatrix} = 0$$

$$\begin{bmatrix} \sum_{j=1}^n a_{1j}x_j \\ \sum_{j=1}^n a_{2j}x_j \\ \vdots \\ \sum_{j=1}^n a_{nj}x_j \end{bmatrix} = 0$$

Since we already assumed A is SDD, and based on the definition $|a_{i,i}| > \sum_{j \neq i} |a_{i,j}|$ it can be deduced that it is required for $a_{ii} \neq 0$ to satisfy the SDD condition. The linear system can be further written in the form

$$\begin{bmatrix} a_{11}x_1 + \sum_{j=2}^n a_{1j}x_j \\ a_{22}x_2 + \sum_{j=1}^{n|n\neq 2} a_{2j}x_j \\ a_{33}x_3 + \sum_{j=1}^{n|n\neq 3} a_{3j}x_j \\ \vdots \\ a_{nn}x_n + \sum_{j=1}^{n-1} a_{nj}x_j \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

This will lead to

$$a_{11}x_1 = -\sum_{j=2}^n a_{1j}x_j$$

$$a_{22}x_2 = -\sum_{j=1}^{n|n\neq 2} a_{2j}x_j$$

$$\ldots$$

$$a_{nn}x_n = -\sum_{j=1}^{n-1} a_{nj}x_j$$

And further

$$\begin{aligned} |a_{11}x_1| &= \left|\sum_{j=2}^{n} a_{1j}x_j\right| \\ |a_{22}x_2| &= \left|\sum_{j=1}^{n|n\neq 2} a_{2j}x_j\right| \\ &\dots \\ |a_{nn}x_n| &= \left|\sum_{j=1}^{n-1} a_{nj}x_j\right| \end{aligned} \tag{1}$$

Or in the simplified form:

$$|a_{ii}x_i| = \left|\sum_{i\neq j} a_{ij}x_j\right|$$

Based on the definition of SDD, we can further derive that:

$$|a_{ii}| > \sum_{i\neq j} |a_{ij}| \geq \left|\sum_{i\neq j} a_{ij}\right|$$

Hence, in order to satisfy $|a_{ii}x_i| = \left|\sum_{i\neq j} a_{ij}x_j\right|$ under the condition of $|a_{ii}| > \left|\sum_{i\neq j} a_{ij}\right|$, is to let $x_k = 0$. In other words, the solution vector $\vec{x}$ has to be

$$\vec{x} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Under this scenario, the linear system $Ax = 0$ has non non-trivial solutions. Hence, if A is SDD, it is also invertible. The statement is proven.

However, in this problem, based on the fact that $|a_{i,i}| > \sum_{j\neq i} |a_{i,j}|$ we already know that the summation of the row[3] shall not be zero. So Equation (1) may not be fully needed to complete the proof. Because based on the fact that row summation shall not be zero discerns that the solution to $\begin{bmatrix} a_{11}x_1 + \sum_{j=2}^{n} a_{1j}x_j \\ a_{22}x_2 + \sum_{j=1}^{n|n\neq 2} a_{2j}x_j \\ \vdots \\ a_{nn}x_n + \sum_{j=1}^{n-1} a_{nj}x_j \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$ should be $\vec{x} = 0$. Hence, both ways complete the proof. □

[3] for any given row

Problem 4. 1. *Compute an LU decomposition of the tridiagonal matrix A by hand, with*

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

Now let $b = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}^T$ and use your computed LU factors to solve the system $Ax = b$ (by hand).

Solution. Given A, Computing the LU decomposition by hand one obtains the following steps:

$$L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ & 1 & 0 & 0 \\ & & 1 & 0 \\ & & & 1 \end{bmatrix}, \quad U = \begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix}$$

$$\Longrightarrow L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1/2 & 1 & 0 & 0 \\ & & 1 & 0 \\ & & & 1 \end{bmatrix}, \quad U = \begin{bmatrix} 2 & -1 & 0 & 0 \\ 0 & 3/2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix}$$

$$\Longrightarrow L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1/2 & 1 & 0 & 0 \\ 0 & -2/3 & 1 & 0 \\ & & & 1 \end{bmatrix}, \quad U = \begin{bmatrix} 2 & -1 & 0 & 0 \\ 0 & 3/2 & -1 & 0 \\ 0 & 0 & 4/3 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix}$$

$$\Longrightarrow L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1/2 & 1 & 0 & 0 \\ 0 & -2/3 & 1 & 0 \\ 0 & 0 & -3/4 & 1 \end{bmatrix}, \quad U = \begin{bmatrix} 2 & -1 & 0 & 0 \\ 0 & 3/2 & -1 & 0 \\ 0 & 0 & 4/3 & -1 \\ 0 & 0 & 0 & 5/4 \end{bmatrix}$$

Verifying the results one may get:

$$LU = \begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix} = A$$

Using the LU factor to solve $Ax = b$:

$$Ax = b$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ -1/2 & 1 & 0 & 0 \\ 0 & -2/3 & 1 & 0 \\ 0 & 0 & -3/4 & 1 \end{bmatrix} \begin{bmatrix} 2 & -1 & 0 & 0 \\ 0 & 3/2 & -1 & 0 \\ 0 & 0 & 4/3 & -1 \\ 0 & 0 & 0 & 5/4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Decomposing to $Ly = b$, one solves

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ -1/2 & 1 & 0 & 0 \\ 0 & -2/3 & 1 & 0 \\ 0 & 0 & -3/4 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$$\to \begin{cases} y_1 = 1 \\ -\frac{1}{2}y_1 + y_2 = 1 \\ -\frac{2}{3}y_2 + y_3 = 1 \\ -\frac{3}{4}y_3 + y_4 = 1 \end{cases} \to \begin{cases} y_1 = 1 \\ y_2 = \frac{3}{2} \\ y_3 = 2 \\ y_4 = \frac{5}{2} \end{cases}$$

One can then solve for $Ux = y$:

$$\begin{bmatrix} 2 & -1 & 0 & 0 \\ 0 & 3/2 & -1 & 0 \\ 0 & 0 & 4/3 & -1 \\ 0 & 0 & 0 & 5/4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 3/2 \\ 2 \\ 5/2 \end{bmatrix}$$

$$\to \begin{cases} 2x_1 - x_2 = 1 \\ \frac{3}{2}x_2 - x_3 = \frac{3}{2} \\ \frac{4}{3}x_3 - x_4 = 2 \\ \frac{5}{4}x_4 = \frac{5}{2} \end{cases} \to \begin{cases} x_1 = 2 \\ x_2 = 3 \\ x_3 = 3 \\ x_4 = 2 \end{cases}$$

The solution vector $\vec{x} = \begin{bmatrix} 2 \\ 3 \\ 3 \\ 2 \end{bmatrix}$ is obtained. □

2. *Using MATLAB, implement the LU decomposition algorithm specialized for tridiagonal matrices. Your code should be able to factor* ***any*** *tridiagonal matrix. Comment on how the computational cost of your algorithm scales with the size of your matrix.*

   Solution. I wrote the following MATLAB function to obtain the LU decomposition for matrix A:

```
function [L,U] = hw2_q4(A)
n = rank(A);
L=eye(n);U=A;
for i=2:n-1
    for j=1:n-2
        if i>j
            L(i,j)=U(i,j)/U(i-1,j);
        end
        if i==j+1
            U(i,j+1)= U(i,j+1) - (U(i,j)*U(i-1,j+1)/U(i-1,j));
        end
        if isnan(L(i,j)) || isnan(U(i,j))
            L(i,j) = 0;U(i,j) = 0;
        end
    end
end

L(n,n-1) = U(n,n-1)/U(n-1,n-1);
U(n,n) = U(n,n) - (L(n,n-1)/L(n-1,n-1)) * U(n-1,n);

for i=2:n
    for j=1:n-1
        if i>j
            U(i,j)=0;
        end
    end
end
fprintf("========================")
end
```

To implement this function, I wrote the following codes:

```
%%
clear;clc
A = [1 -8 0 0; 2 -2 -7 0; 0 7 3 -6; 0 0 8 -7];
[L_test,U_test] = hw2_q4(A);
err1 = A-L_test*U_test
%%
clear;
A = [9 2 0 0 0; 3 5 -2 0 0; 0 2 8 -6 0;0 0 3 9 -7; 0 0 0 1 5];
[L_test,U_test] = hw2_q4(A);
err2 = A-L_test*U_test
%%
clear;
A = [9 2 0 0 0 0; 3 5 -2 0 0 0; 0 2 8 -6 0 0;0 0 3 9 -7 0; 0 0 0 1 5 0; 0 0 0 0 3 2];
[L_test,U_test] = hw2_q4(A);
err3 = A-L_test*U_test
```

and the corresponding three errors are shown as:

```
========================
err1 =

     0     0     0     0
     0     0     0     0
     0     0     0     0
     0     0     0     0

========================
err2 =

   1.0e-15 *

         0         0         0         0         0
         0         0         0         0         0
         0         0         0         0         0
         0         0         0         0         0
         0         0         0    0.1110         0
```

```

========================
err3 =

   1.0e-15 *

         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    0.1110         0         0
         0         0         0         0    0.4441         0
```

Indicating the algorithm works, with acceptable errors ($< 10^{-15}$).

In my code implementation, I used two "`for`" loops to assign the updated values to matrices L and U. Assume the dimension of the matrix is d. Hence, my algorithm scales the square relationship to the matrix size, i.e. $\mathcal{O}(d^2)$.[4] However, based on Prof. Gerristen's note, I realize this LU decomposition can also be achieved in just one `for` loop, in that case, the computational complexity is $\mathcal{O}(d)$. Hence, my algorithm is definitely not the most efficient way to conduct LU decomposition for a given A, but it can achieve the objective with acceptable accuracy. In my algorithm implementation, for example, when the matrix size increases from 4 to 5, the computational burden increases scaling is approximately $\frac{25}{16}$.$\square$

[4] because after expansion the lower-order terms of d can be ignored, hence the overall computational complexity is still d.

Problem 5. *We are interested in solving the 1D heat equation numerically. In 1D, the heat equation has the form*

$$\frac{d^2T}{dx^2} = f(x), \text{ for } 0 \le x \le 1,$$

with x denoting the distance along a rod with constant thermal conductivity, T denoting the temperature of the rod, and f denoting the distributed heat source.

Discretize the equation using the second-order central finite difference scheme on a uniform grid with spacing $h = 1/N$ (see Section 1.7 in Prof. Gerritsen's note for a derivation).

Consider the source term $f(x) = -10\sin\left(\frac{3\pi x}{2}\right)$ and fix the boundary conditions $T(0) = 0$ and $T(1) = 2$.

1. *Verify that*

$$T_{\text{exact}}(x) = \left(2 + \frac{40}{9\pi^2}\right)x + \frac{40}{9\pi^2}\sin\left(\frac{3\pi x}{2}\right)$$

is the exact solution to the heat equation with the given source term and boundary conditions.

Solution. Solving the heat equation using the given source term and boundary conditions, one has

$$\begin{aligned} T &= \int\int f(x)dxdx \\ &= \int\int \left[-10\sin\left(\frac{3\pi x}{2}\right)\right] dxdx \\ &= \int \left[\frac{20}{3\pi}\cos\left(\frac{3\pi x}{2}\right) + c\right] dx \\ &= \frac{40}{9\pi^2}\sin\left(\frac{3\pi x}{2}\right) + cx \end{aligned}$$

SUbstituting the boundary conditions $T(0) = 0$ and $T(1) = 2$, one has

$$\begin{aligned} -\frac{40}{9\pi^2} + c = 2 \\ c = 2 + \frac{40}{9\pi^2} \end{aligned}$$

One hence obtain the analytical solution:

$$T(x) = \left(2 + \frac{40}{9\pi^2}\right)x + \frac{40}{9\pi^2}\sin\left(\frac{3\pi x}{2}\right)$$

The solved T matched with the exact solution T_{exact}. The given relationship is hence proved. □

2. *Use your specialized tridiagonal LU implementation from Problem 4 to obtain a numerical approximation for $T_j = T(jh)$, $j = 1, \ldots, N-1$, for $N = 10, 20, 40, 80, 160$. Plot all your approximations together with the exact solution on the same set of axes. Comment on the relationship between N and the approximation error $\|T_{\text{numerical}} - T_{\text{exact}}\|$.*

Solution. Using the second-order central-difference scheme, we have

$$\frac{d^2T(x_i)}{dx^2} \approx \frac{T_{i+1} - 2T_i + T_{i-1}}{h^2}$$

Referring to section 1.7 in Prof. Gerritsen's note, by plugging in the approximation, one has

$$T_{i+1} - 2T_i + T_{i-1} = h^2 f_i$$

Given the boundary conditions, $T(0) = 0$ and $T(1) = 2$, one can reformulate the finite difference approximation scheme into $A\vec{T} = \vec{c}$, where the matrix can be expanded as

$$A = \begin{bmatrix} -2 & 1 & & & & & \\ 1 & -2 & 1 & & & & \\ & 1 & -2 & 1 & & & \\ & & \vdots & & & & \\ & & 1 & -2 & 1 & & \\ & & & 1 & -2 & 1 & \\ & & & & 1 & -2 & 1 \end{bmatrix}, \quad \vec{T} = \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ \vdots \\ T_{N-3} \\ T_{N-2} \\ T_{N-1} \end{bmatrix}, \quad \vec{c} = \begin{bmatrix} h^2 f_1 \\ h^2 f_2 \\ h^2 f_3 \\ \vdots \\ h^2 f_{N-3} \\ h^2 f_{N-2} \\ h^2 f_{N-1} - 2 \end{bmatrix}$$

Using the LU decomposition obtained from Q4, we can rewrite the system into

$$[L]\,[U] \begin{bmatrix} T_1 \\ T_2 \\ \cdots \\ T_N \end{bmatrix} = \begin{bmatrix} h^2 f_1 \\ h^2 f_2 \\ \cdots \\ h^2 f_{N-1} - 2 \end{bmatrix}$$

Based on this simple formulation, we can write a MATLAB function to obtain the numerical solution $\vec{T}$ given different grid size N:

```matlab
function [T_vec, T_exact, error] = hw2_q5(N)
x = 0:1/(N-1):1;%define grid
f = -10*sin((3*pi*x)/2);%define source term
h = 1/N;
T_vec = ones(N,1);
c_vec = h^2*f';
c_vec(end) = c_vec(end)-2;
for i = 1:N-1
    for j = i:N-1
        if i==j
            A(i,j)=-2;
            A(i,j+1)=1;
            A(i+1,j)=1;
        end
    end
end
A(N,N)=-2;
[L,U]=hw2_q4(A);
y_vec = L\c_vec;
T_vec = U\y_vec;% numerically approximated
```

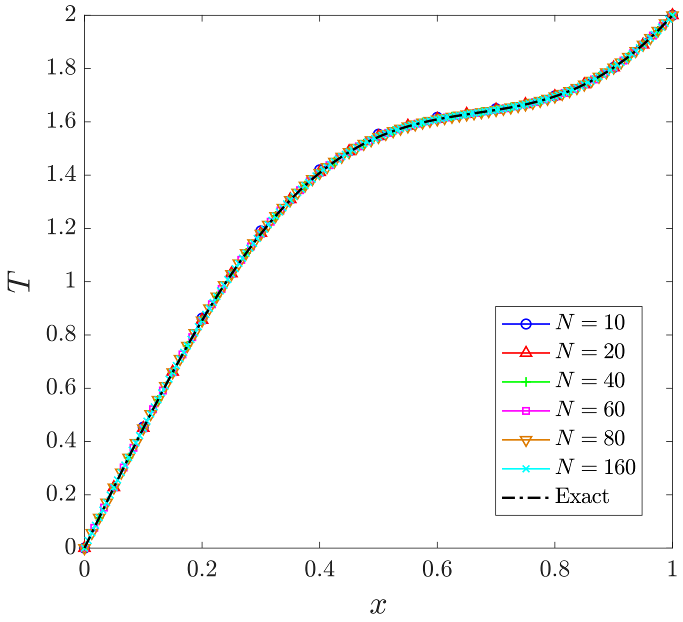


Figure 1: Numerical approximation solutions comparison.

```
%% exact solution
T_exact = (2 + (40/(9*pi^2)))*x + ( 40/(9*pi^2) )*sin( (3*pi*x)/2 );
error = norm(T_vec-T_exact);
end
```

One obtains Figure 1 by plotting the numerically approximated solutions with the exact solution. For a better understanding of the approximation error, we also plot the norms $\|T_{nuemrical} - T_{exact}\|$ in Figure 2, by recalling the MATLAB "`norm()`" function[5]. It is observed that the norm decreases in an exponential fashion.

Theoretically, with more data points corresponding to the increasing grid number, one may expect the cumulative L2 norm to increase as the evaluated data points increase. But simultaneously the error between the exact solution and the approximations also decreases. Figure 1 shows that the decreasing trend of the difference between the approximation and the exact solution plays a dominant role for the L2 norm whereas the increasing data point effect is hence relatively low.

□

[5] which is the Euclidean norm, or also known as the L2 norm

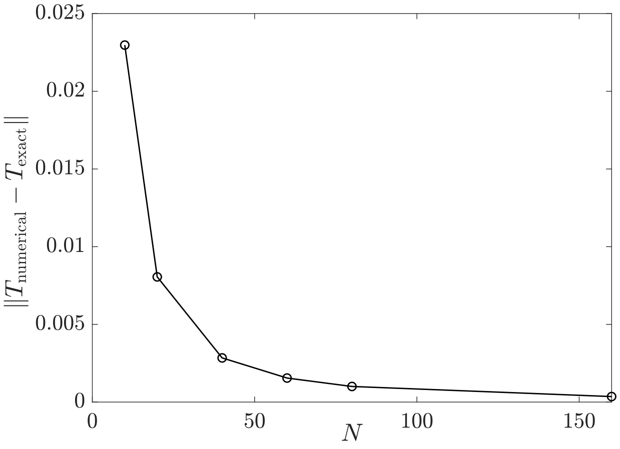


Figure 2: The norm for the central difference scheme with different N.

ME300A HW#3
Hanfeng Zhai (hzhai@stanford.edu) November 7, 2023

Problem 1. *This problem explores issues that arise when computing the QR factorization numerically. In lecture we explained how to use the Gram-Schmidt procedure to construct an orthonormal basis of the column space of a given matrix. The problem is that, in numerical computations, the vectors produced by the Gram-Schmidt recipe gradually lose orthogonality. See this for yourself!*

(a) *Let M denote the $n \times n$ Hilbert matrix, with entries $m_{ij} = \frac{1}{i+j-1}$. Set $n = 15$ and use the Gram-Schmidt procedure to find $Q = \begin{bmatrix} q_1 & \cdots & q_n \end{bmatrix}$.*

The theory tells us Q should be orthogonal so that $Q^\mathsf{T} Q = I$. Test this by computing `norm(Q' * Q - eye(15))`. Report the norm you found and briefly comment on your result: does this computation agree with the theory we discussed?

***Note:** The built-in `qr` function in MATLAB performs more sophisticated calculations, so you will have to implement your own Gram-Schmidt routine.*

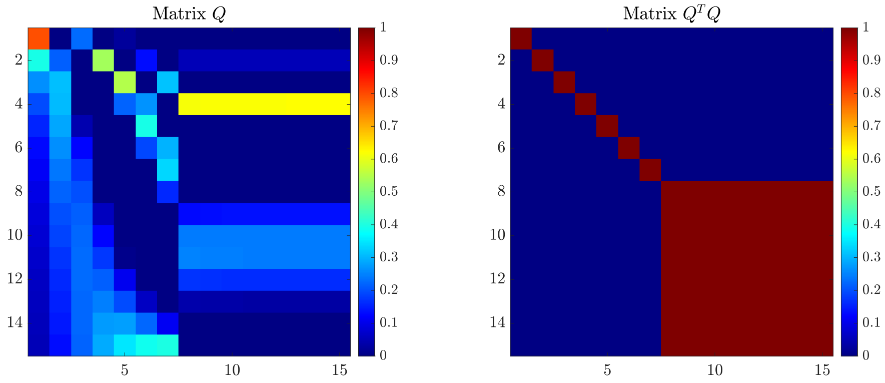
Solution.

In this problem, I have two approaches that give me slightly different results. The second one reports slightly more accurate $Q^\mathsf{T} Q$ results but does not follow the standard solution procedure. I will report both of them here.

For my first approach, I follow the standard textbook formula, containing codes as follows

```matlab
M = zeros(15, 15);
for i = 1:15
  for j = 1:15
    M(i, j) = 1 / (i + j - 1);
  end
end
for j = 1:15
    v = M(:,j);
    for i = 1:j-1
        R(i,j) = Q(:,i)'*M(:,j);
        v= v-R(i,j)*Q(:,i);
    end
    R(j,j) = norm(v);
    Q(:,j)=v/R(j,j);
end
norm = norm(Q' * Q - eye(15));
```

By using this code, the reported Q and $Q^\mathsf{T} Q$ are shown in the following figure.

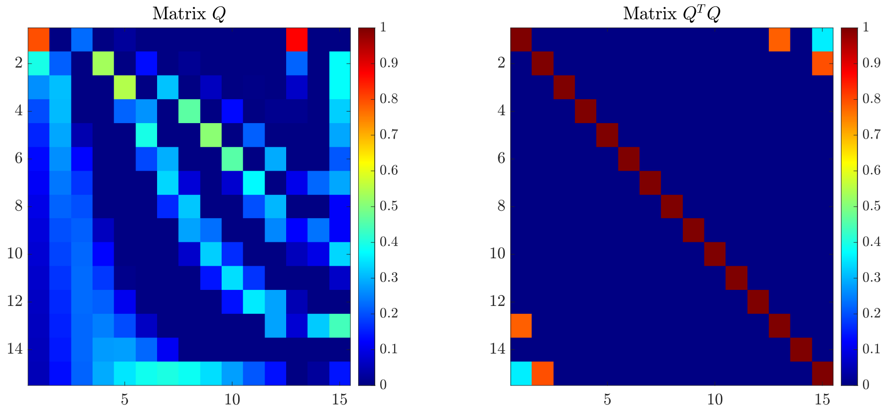


It can be observed that the error starts to propagate after column 7, and the reported $Q^\mathsf{T}Q$ is very inaccurate. In this case, the reported norm (`norm = norm(Q' * Q - eye(15))`) is **7.9351**. And obviously, this does not agree with the theory we discussed.

Following this result, I was not satisfied with the accuracy. There is another approach that does not follow the standard procedure: instead of using the MATLAB multiplication "`*`", I used the dot product "`dot()`", and surprisingly the results improved quite a bit! Hence I also report that approach here. My MATLAB codes write:

```
clc;clear;close all
M = zeros(15, 15);
for i = 1:15
  for j = 1:15
    M(i, j) = 1 / (i + j - 1);
  end
end

Q = zeros(15, 15);
for i = 1:15
  v = M(:, i);
  for j = 1:i - 1
    v = v - Q(:, j) * dot(Q(:, j), v);
  end
  Q(:, i) = v / norm(v); sum(Q(:, i));
end
norm = norm(Q' * Q - eye(15));
```

Using this code one can also plot the matrices for both M, Q, and $Q^\mathsf{T}Q$, shown as follows:



It can be seen that this approach indeed improves the accuracy, even though it does not follow the standard solution procedure. The calculated corresponding norm is **0.9961**, indicating there are some system-related numerical errors involved during the QR decomposition process, but it gives the generally accurate Q matrix.

In short, the first method reported follows the standard QR decomposition, yet reports a pretty high norm. The second method is more of a personal way to tweak for more accurate results. Both methods are not accurate based on the evaluated norms. □

Householder matrices arose as a solution to this problem. The Householder reflection H_v is defined by

$$H_v = I_n - \frac{2}{v^\intercal v} v v^\intercal.$$

We now turn to studying some properties of H_v. These will help us understand how to use Householder reflections to develop a numerically stable QR factorization.

(b) *Show that H_v is symmetric and orthogonal.*

Solution. In this problem, one needs to prove

$$\begin{cases} H_v^\intercal = H_v \\ H_v H_v^\intercal = I \end{cases} \tag{1}$$

One can begin with writing out $H_v^\intercal$:

$$\begin{aligned} H_v^\intercal &= \left(I_n - \frac{2}{v^\intercal v} v v^\intercal \right)^\intercal \\ &= I_n^\intercal - \left(\frac{2}{v^\intercal v} v v^\intercal \right)^\intercal \\ &= I_n^\intercal - (v v^\intercal)^\intercal \left(\frac{2}{v^\intercal v} \right)^\intercal \\ &= I_n - 2 v v^\intercal \frac{1}{v^\intercal v} \\ &= I_n - \frac{2}{v^\intercal v} v v^\intercal = H_v \end{aligned} \tag{2}$$

The matrix H_v is hence proved to be symmetric. To show they are orthogonal, we can then expand $H_v^\top H_v$:

$$\begin{aligned} H_v^\top H_v &= \left(I_n - vv^\top \frac{2}{v^\top v}\right)\left(I_n - \frac{2}{v^\top v} vv^\top\right) \\ &= \left(I_n - vv^\top \frac{2}{v^\top v}\right)^2 \\ &= I_n - 4\frac{vv^\top}{v^\top v} + 4\frac{vv^\top vv^\top}{v^\top vv^\top v} \\ &= I_n - 4\frac{vv^\top}{v^\top v} + \frac{4vv^\top}{v^\top v} \\ &= I_n - 4\frac{vv^\top}{v^\top v} + 4\frac{vv^\top}{v^\top v} \\ &= I_n \end{aligned} \tag{3}$$

The matrix H_v are hence probed to be orthogonal. □

(c) *Show that $H_v v = -v$. Also show that if w is orthogonal to v, then $H_v w = w$.*

Solution. First, to show $H_v v = -v$, we begin with expanding H_v:

$$\begin{aligned} H_v v &= \left(I_n - \frac{2}{v^\top v} vv^\top\right) v \\ &= v - \frac{2}{v^\top v} vv^\top v \\ &= v - 2v \\ &= -v \end{aligned} \tag{4}$$

Now, take the assumption of w is orthogonal to v, we know that $\vec{w}^\top \vec{v} = 0$. We can further expand $H_v w = w$:

$$\begin{aligned} H_v w &= \left(I_n - \frac{2}{v^\top v} vv^\top\right) w \\ &= w - \frac{2}{v^\top v} vv^\top w \end{aligned} \tag{5}$$

Since we already know that $\vec{v}^\top \vec{w} = \vec{w}^\top \vec{v} = 0$. We further expand Eqn. 5:

$$H_v w = w \tag{6}$$

The statement is hence proved. □

(d) *Now suppose u and w are vectors such that $\|u\| = \|w\|$. Show that $H_{u-w} u = w$.*

Hint: Write $u = \frac{1}{2}\big((u - w) + (u + w)\big)$, show that $(u - w)^\top (u + w) = 0$, and consider your previous results.

Solution. Based on the hint, we may begin with trying to prove

$$(u-w)^\mathsf{T}(u+w)=0 \tag{7}$$

Since $\|u\|=\|w\|$, we may further expand the $(u-w)^\mathsf{T}(u+w)$:

$$(u-w)^\mathsf{T}(u+w)=u^\mathsf{T}u+u^\mathsf{T}w-w^\mathsf{T}u-w^\mathsf{T}w \tag{8}$$

One may assume the contact angle u and w is θ. Hence:

$$\begin{aligned} u^\mathsf{T}w &= \|u\|\|w\|\cos\theta \\ w^\mathsf{T}u &= \|w\|\|u\|\cos\theta \end{aligned} \tag{9}$$

We may substitute back to the previous equation, getting:

$$\begin{aligned} u^\mathsf{T}u+u^\mathsf{T}w-w^\mathsf{T}u-w^\mathsf{T}w &= \underbrace{\|u\|\|u\|-\|w\|\|w\|}_{=0}+\underbrace{\|u\|\|w\|\cos\theta-\|w\|\|u\|\cos\theta}_{=0} \\ &= 0 \end{aligned} \tag{10}$$

This equation is hence proved.

Based on the results in (c), one has

$$\begin{aligned} H_{u-w}(u-w) &= w-u \\ H_{u-w}u-H_{u-w}w &= w-u \\ H_{u-w}u &= \underbrace{H_{u-w}w}_{\text{expand}}+w-u \end{aligned} \tag{11}$$

By expanding the marked term we have:

$$\begin{aligned} H_{u-w}w &= \left(I_n-\frac{2}{(u-w)^\mathsf{T}(u-w)}(u-w)(u-w)^\mathsf{T}\right)w \\ &= w-2\frac{uu^\mathsf{T}w-uw^\mathsf{T}w-wu^\mathsf{T}w+ww^\mathsf{T}w}{u^\mathsf{T}u-u^\mathsf{T}w-w^\mathsf{T}u+w^\mathsf{T}w} \\ &= \frac{wu^\mathsf{T}u-wu^\mathsf{T}w-ww^\mathsf{T}u+ww^\mathsf{T}w-2(uu^\mathsf{T}w-uw^\mathsf{T}w-wu^\mathsf{T}w+ww^\mathsf{T}w)}{u^\mathsf{T}u-u^\mathsf{T}w-w^\mathsf{T}u+w^\mathsf{T}w} \\ &= \frac{wu^\mathsf{T}(u+w)-ww^\mathsf{T}(u+w)-2uu^\mathsf{T}w+2uw^\mathsf{T}w}{u^\mathsf{T}u-u^\mathsf{T}w-w^\mathsf{T}u+w^\mathsf{T}w} \\ &= \frac{(wu^\mathsf{T}-ww^\mathsf{T})(u+w)+2(uw^\mathsf{T}-uu^\mathsf{T})w}{u^\mathsf{T}u-u^\mathsf{T}w-w^\mathsf{T}u+w^\mathsf{T}w} \\ &= \frac{\overbrace{w(u^\mathsf{T}-w^\mathsf{T})(u+w)}^{=0}+2u(w^\mathsf{T}-u^\mathsf{T})w}{(u-w)^\mathsf{T}(u-w)} \end{aligned} \tag{12}$$

Now, we may substitute Eqn. (12) back to Eqn. (11):

$$\begin{aligned}
H_{u-w}w &= \frac{2u(w^\top - u^\top)w + (w-u)(u^\top - w^\top)(u-w)}{(u-w)^\top(u-w)} \\
&= \frac{(2uw^\top - 2uu^\top)w + (wu^\top - ww^\top - uu^\top + uw^\top)(u-w)}{(u-w)^\top(u-w)} \\
&= \frac{uw^\top(w+u) - uu^\top(w+u) + wu^\top(u-w) + ww^\top(w-u)}{(u-w)^\top(u-w)} \\
&= \frac{(uw^\top - uu^\top)(u+w) + (wu^\top - ww^\top)(u-w)}{(u-w)^\top(u-w)} \\
&= \frac{\overbrace{u(w-u)^\top(u+w)}^{=0} + w(u-w)^\top(u-w)}{(u-w)^\top(u-w)} \\
&= w
\end{aligned} \tag{13}$$

The statement is hence proved. □

(e) *Consider the matrix*

$$A = \begin{bmatrix} 2 & 3 \\ -2 & 4 \\ 1 & 5 \end{bmatrix}.$$

Much like an elementary row matrix, H_v can be used to zero out elements in a column of A when v is chosen appropriately.

Let a_1 denote the first column of A. Find a vector $v \in \mathbb{R}^3$ such that

$$H_v a_1 = \|a_1\| \, e_1,$$

where $e_1 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^\top$. Report v and also the product $H_v A$.

Hint: Consider the result of part (d).

Solution. Taking the hint, since in (d) the vector is formulated as $v = u - w$. Here, we take a similar approach by setting $v = a_1 - \|a_1\|e_1$. To verify this, the equation writes:

$$\begin{aligned}
H_{a_1-\|a_1\|e_1} a_1 &= \|a_1\|e_1 \\
I_n - \frac{2(a_1 - \|a_1\|e_1)(a_1 - \|a_1\|e_1)^\top}{(a_1 - \|a_1\|e_1)^\top(a_1 - \|a_1\|e_1)} a_1 &= \|a_1\|e_1 \\
I_n - \frac{2(a_1 - \|a_1\|e_1)(a_1^\top - \|a_1\|e_1^\top)}{(a_1^\top - \|a_1\|e_1^\top)(a_1 - \|a_1\|e_1)} &= \frac{\|a_1\|e_1}{a_1}
\end{aligned} \tag{14}$$

By expanding the left-hand side one has:

$$I_n - \frac{2(a_1 a_1^\top - a_1\|a_1\|e_1^\top - \|a_1\|e_1 a_1^\top + \|a_1\|^2 e_1 e_1^\top)}{a_1^\top a_1 - a_1^\top\|a_1\|e_1 - \|a_1\|e_1^\top a_1 - \|a_1\|^2 e_1^\top e_1} = \frac{\|a_1\|e_1}{a_1} \tag{15}$$

The equation is hence established. Therefore the vector $v = a_1 - \|a_1\|e_1$ satisfy the condition. □

(f) *The result of (e) suggests how to compute Q using Householder reflections: at the kth step, choose v_k appropriately to zero out the kth column of A below the diagonal. Applying the corresponding Householder reflections successively, we obtain an upper triangular matrix R:*

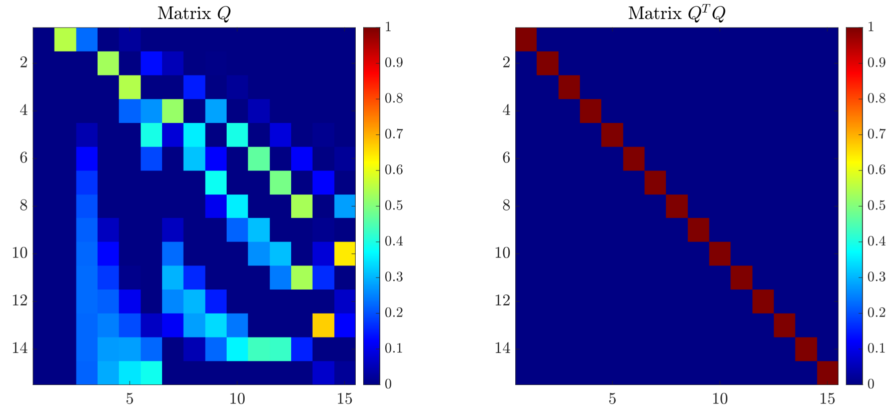
$$H_{v_{n-1}} \cdots H_{v_1} A = R.$$

Thus we obtain $A = QR$ by setting $Q = H_{v_1} \cdots H_{v_n}$, since each reflection is symmetric and orthogonal.

Implement this procedure in MATLAB. Obtain an orthonormal basis for the column space of the Hilbert matrix M and report `norm(Q' * Q - eye(15))` *in this case.*

Solution. Given the provided instructions, we can write the new QR decomposition for M (by setting the Hilbert matrix, i.e., MATLAB `hilb()`, from the instructions):

```
n = 15;
M = hilb(n);

Q = eye(n);
for k = 1:n-1
    x = M(k:n, k);
    v = x;
    v(1) = v(1) + sign(x(1)) * norm(x); % Choose appropriate v_k
    v = v / norm(v);

    H = eye(n);
    H(k:n, k:n) = H(k:n, k:n) - 2 * (v * v');
    M = H * M;
    Q = Q * H';
end

R = M;
norm = norm(Q' * Q - eye(15));
```



The reported `norm(Q' * Q - eye(15))` is 1.76×10^{-15}. It can be deduced from both the error and the matrix visualization that QR decomposition using this method is much more accurate than that of what I wrote in (a).

The reported Q matrix (orthonormal basis) is

```
Q =

   Columns 1 through 9

   -0.7954    0.5546    0.2297   -0.0792    0.0242   -0.0067
      -0.0017    0.0004   -0.0001
   -0.3977   -0.2187   -0.6290    0.5333   -0.3040    0.1364
      0.0511   -0.0165    0.0046
   -0.2651   -0.3111   -0.3205   -0.2271    0.5476   -0.5031
      -0.3141    0.1517   -0.0597
   -0.1989   -0.3077   -0.0918   -0.3926    0.2201    0.2673
      0.5198   -0.4617    0.2826
   -0.1591   -0.2858    0.0454   -0.3396   -0.1129    0.3963
      0.0830    0.3566   -0.5090
   -0.1326   -0.2618    0.1263   -0.2297   -0.2787    0.1933
      -0.2967    0.3137    0.1231
   -0.1136   -0.2396    0.1739   -0.1176   -0.3139   -0.0501
      -0.3355   -0.0828    0.3840
   -0.0994   -0.2200    0.2017   -0.0197   -0.2717   -0.2186
      -0.1631   -0.3184    0.1061
   -0.0884   -0.2029    0.2173    0.0611   -0.1898   -0.2898
      0.0588   -0.2820   -0.2343
   -0.0795   -0.1880    0.2253    0.1260   -0.0916   -0.2762
      0.2276   -0.0734   -0.3245
   -0.0723   -0.1750    0.2285    0.1774    0.0097   -0.1994
      0.2969    0.1615   -0.1435
   -0.0663   -0.1636    0.2285    0.2178    0.1072   -0.0799
      0.2566    0.3033    0.1497
   -0.0612   -0.1535    0.2265    0.2493    0.1974    0.0656
      0.1154    0.2778    0.3381
   -0.0568   -0.1446    0.2232    0.2738    0.2787    0.2246
      -0.1097    0.0507    0.2234
   -0.0530   -0.1366    0.2191    0.2926    0.3511    0.3884
      -0.3998   -0.3833   -0.3410

   Columns 10 through 15

    0.0000   -0.0000    0.0000   -0.0000    0.0000    0.0000
   -0.0011    0.0002   -0.0000    0.0000   -0.0000   -0.0000
    0.0196   -0.0054    0.0012   -0.0002    0.0001   -0.0000
   -0.1323    0.0492   -0.0147    0.0032   -0.0017    0.0002
    0.3966   -0.2163    0.0882   -0.0258    0.0128   -0.0027
   -0.4602    0.4651   -0.2878    0.1197   -0.0528    0.0204
   -0.0776   -0.3698    0.4896   -0.3334    0.1229   -0.0944
    0.3556   -0.2028   -0.2952    0.5367   -0.1393    0.2786
    0.2195    0.3114   -0.2549   -0.3844    0.0119   -0.5330
   -0.1676    0.2644    0.3097   -0.1715    0.0852    0.6434
   -0.3404   -0.1724    0.2439    0.5353    0.1649   -0.4324
   -0.1386   -0.3546   -0.2809   -0.2937   -0.5970    0.0672
    0.2403    0.0015   -0.2866   -0.0971    0.6684    0.1226
    0.3653    0.4372    0.4263    0.1560   -0.3460   -0.0899
   -0.2792   -0.2077   -0.1388   -0.0449    0.0706    0.0200
```

□

Problem 2. *(Spanning set, basis.) Consider the following vectors in* $\mathbb{R}^3$*:*

$$\vec{v_1} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}, \quad \vec{v_2} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \vec{v_3} = \begin{bmatrix} 3 \\ 5 \\ 4 \end{bmatrix}, \quad \vec{v_4} = \begin{bmatrix} 4 \\ 0 \\ 4 \end{bmatrix}, \quad \textit{and} \quad \vec{v_5} = \begin{bmatrix} 0 \\ 2 \\ 1 \end{bmatrix}.$$

(a) *Is this a spanning set for* $\mathbb{R}^3$*? Why?*

Solution. For this problem, we can construct a matrix containing the vector sets:

$$V = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 1 & 0 \\ 3 & 5 & 4 \\ 4 & 0 & 4 \\ 0 & 2 & 1 \end{bmatrix} \tag{16}$$

By conducting the Gaussian elimination (or using `rref` in MATLAB) one can obtain its reduced echelon form:

$$V_r = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{17}$$

One can clearly observe from the reduced echelon form that the rank of the matrix is 3. Hence, it spans $\mathbb{R}^3$.□

(b) *Prove that* $v_1, \ldots, v_5$ *are linearly dependent. Reduce the list to a basis of* $\mathbb{R}^3$ *by removing redundant vectors.*

Solution. To the prove the five vectors are linearly dependent, we may construct the a constant vector $\vec{\alpha} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \end{bmatrix}$ such that

$$\alpha_1\vec{v_1} + \alpha_2\vec{v_2} + \alpha_3\vec{v_3} + \alpha_4\vec{v_4} + \alpha_5\vec{v_5} = 0 \tag{18}$$

or in the matrix form

$$\mathcal{V}\vec{\alpha} = 0 \quad \Longrightarrow \quad \begin{bmatrix} 1 & 0 & 3 & 4 & 0 \\ 2 & 1 & 5 & 0 & 2 \\ 1 & 0 & 4 & 4 & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \end{bmatrix} = 0 \tag{19}$$

It can be seen that there are only three constraints yet five unknowns. Hence, there will be infinitely amount of solutions exist. Therefore, the five vectors are linearly dependent.

To reduce the list to $\mathbb{R}^3$ basis, we can calculate the reduced echelon form of this coefficient matrix:

$$\mathcal{V}_r = \begin{bmatrix} 1 & 0 & 0 & 4 & -3 \\ 0 & 1 & 0 & -8 & 3 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix} \tag{20}$$

It can be deduced from $\mathcal{V}_r$ that only the first three column vectors are linearly independent. Hence, the reduced list writes:

$$\left\{ \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 3 \\ 5 \\ 4 \end{bmatrix} \right\} \tag{21}$$

□

(c) *Express one of the redundant vectors as a linear combination of the basis you found in (b).*

Solution. It can be identified that the fourth vector can be written as the linear combination of the basis in the form of

$$4 \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} - 8 \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + 0 \begin{bmatrix} 3 \\ 5 \\ 4 \end{bmatrix} = \begin{bmatrix} 4 \\ 0 \\ 4 \end{bmatrix} \tag{22}$$

□

Problem 3. *(Column space, row space, null space.) Consider the following matrix A.*

$$A = \begin{bmatrix} 3 & 4 & -1 & 15 & 12 \\ 2 & 2 & 4 & -10 & -12 \\ 1 & 1 & 2 & -5 & 3 \\ -2 & -3 & 3 & -20 & -18 \end{bmatrix}$$

(a) *Find the condition(s) on an arbitrary vector $\vec{b}$ such that $A\vec{x} = \vec{b}$ has at least one solution. Is the solution unique? Why?*

Solution. We can first calculate the reduced echelon form of A:

$$A_r = \begin{bmatrix} 1 & 0 & 9 & -35 & 0 \\ 0 & 1 & -7 & 30 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{23}$$

It can be observed that A is not fully ranked. Hence, in order for $A\vec{x} = \vec{b}$ to have at least one solution, $\vec{b}$ has to lie in the column space of A. The solution is not unique, since A is not full-ranked. There will be an infinite set of solutions.

Here, one also needs to identify the linear combination between the row spaces of A: $\alpha_1\vec{a_1} + \alpha_2\vec{a_2} + \alpha_3\vec{a_3} + \alpha_4\vec{a_4} = 0$. One can then solve that

$$\begin{cases} \alpha_1 = -2 \\ \alpha_2 = 1 \\ \alpha_3 = 0 \\ \alpha_4 = 2 \end{cases} \tag{24}$$

So the relationship for vector b is $-2b_1 + b_2 - 2b_4 = 0$ □

(b) *Find the rank of A and provide a basis for the row space of A.*

Solution. Based on the reduced echelon form given in (a), one knows the rank is 3. From the reduced echelon form, we can also identify the basis for the row space as the first three row-vectors:

$$\mathcal{B}_{row} = \left\{ \begin{bmatrix} 3 \\ 4 \\ -1 \\ 15 \\ 12 \end{bmatrix}, \begin{bmatrix} 2 \\ 2 \\ 4 \\ -10 \\ -12 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 2 \\ -5 \\ 3 \end{bmatrix} \right\} \tag{25}$$

□

(c) *Find a basis for the null space of A. What is its dimension?*

Solution. From the reduced echelon form, we can write out the null space (solutions for $A\vec{x} = 0$)as the form of linear combinations of x_i:

$$\begin{aligned} x_1 + 9x_3 - 35x_4 &= 0 \\ x_2 - 7x_3 + 30x_4 &= 0 \\ x_5 &= 0 \end{aligned} \tag{26}$$

one further deduce:

$$\begin{aligned} x_1 &= 35x_4 - 9x_3 \\ x_2 &= 7x_3 - 30x_4 \end{aligned} \tag{27}$$

One can then write out the form of the basis by setting $x_3 \to t$ and $x_4 \to s$:

$$\begin{aligned} x_1 &= -9t + 35s \\ x_2 &= 7t - 30s \\ x_3 &= t \\ x_4 &= s \\ x_5 &= 0 \end{aligned} \tag{28}$$

The basis can then be expanded in the form of

$$\begin{bmatrix} -9 \\ 7 \\ 1 \\ 0 \\ 0 \end{bmatrix} t + \begin{bmatrix} 35 \\ -30 \\ 0 \\ 1 \\ 0 \end{bmatrix} s \tag{29}$$

Or in the form of a set

$$\mathcal{B}_{null} = \left\{ \begin{bmatrix} -9 \\ 7 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 35 \\ -30 \\ 0 \\ 1 \\ 0 \end{bmatrix} \right\} \tag{30}$$

where the vectors in the nullspace of A are linear combinations of the two vectors. □

(d) *Verify that every vector in $\mathcal{N}(A)$ is orthogonal to every vector in* row(A).

Solution. To verify this, we can begin with constructing two matrices, one consists of the basis and the general form of their linear combinations (denoted as $N_{\mathcal{B}}$), and the other consists of the row vectors in A. Once the multiplication result is a zero matrix, one can hence prove that all the vectors in $\mathcal{N}(A)$ are orthogonal to every vector in row(A). One hence write out the two matrices $N_{\mathcal{B}}^{\mathsf{T}} A^{\mathsf{T}}$:

$$\begin{bmatrix} -9 & 7 & 1 & 0 & 0 \\ 35 & -30 & 0 & 1 & 0 \\ 35s - 9t & 7t - 30s & t & s & 0 \end{bmatrix} \begin{bmatrix} 3 & 2 & 1 & -2 \\ 4 & 2 & 1 & -3 \\ -1 & 4 & 2 & 3 \\ 15 & -10 & -5 & -20 \\ 12 & -12 & 3 & -18 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{31}$$

This verifies that every vector in $\mathcal{N}(A)$ is orthogonal to every vector in $\text{row}(A)$. One can also verify it using MATLAB:

```
syms t s

b1 = [-9 7 1 0 0]';
b2 = [35 -30 0 1 0]';
B_null = [b1, b2, b1*t+b2*s];

A = [3 4 -1 15 12;...
    2 2 4 -10 -12;...
    1 1 2 -5 3;...
    -2 -3 3 -20 -18];
A_row = A';
B_null'*A_row
```

and the corresponding output is

```
ans =

[0, 0, 0, 0]
[0, 0, 0, 0]
[0, 0, 0, 0]
```

□

(e) *Find the dimension and basis for the column space of A.*

Solution. Recall the reduced echelon form of A we find in (a):

$$A_r = \begin{bmatrix} 1 & 0 & 9 & -35 & 0 \\ 0 & 1 & -7 & 30 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{32}$$

One can determine the basis for the column space of A:

$$\mathcal{B}_{col} = \left\{ \begin{bmatrix} 3 \\ 2 \\ 1 \\ -2 \end{bmatrix}, \begin{bmatrix} 4 \\ 2 \\ 1 \\ -3 \end{bmatrix}, \begin{bmatrix} 12 \\ -12 \\ 3 \\ -18 \end{bmatrix} \right\} \tag{33}$$

The dimension is then 3. □

Problem 4. *(Properties of Determinants.)*

(a) *Prove that the determinant of an orthogonal matrix is either* $+1$ *or* -1.

Solution. One begins with defining an orthogonal matrix Q, preserving the property:

$$Q^{\mathsf{T}}Q = QQ^{\mathsf{T}} = I \tag{34}$$

Using the property of determinants we have

$$\begin{aligned} \det\left(Q^{\mathsf{T}}Q\right) &= \det\left(Q^{\mathsf{T}}\right)\det\left(Q\right) \\ &= \det\left(I\right) = 1 \end{aligned} \tag{35}$$

One further writes:

$$\left[\det\left(Q\right)\right]^2 = 1 \tag{36}$$

We can then conclude that

$$\det\left(Q\right) = \pm 1 \tag{37}$$

The statement is hence proved. □

(b) *Suppose* L *is an* $n \times n$ *lower triangular matrix. Show that* $\det(L)$ *is the product of the diagonal entries of* L*; that is, prove that*

$$\det(L) = \ell_{11} \cdots \ell_{nn}.$$

Solution. One way to prove this is to expand the terms in L:

$$L = \begin{bmatrix} \ell_{11} & 0 & 0 & \dots & 0 \\ \ell_{21} & \ell_{22} & 0 & \dots & 0 \\ \ell_{31} & \ell_{32} & \ell_{33} & \dots & 0 \\ \vdots & & & \ddots & \vdots \\ \ell_{n1} & \ell_{n2} & \dots & & \ell_{nn} \end{bmatrix} \tag{38}$$

By computing the determinant one has

$$\begin{aligned} \det\left(L\right) &= \ell_{11}\det\left(L_{22}\right) \\ &= \ell_{11}\det\left(\ell_{22}\det\left(L_{33}\right)\right) \\ &= \ell_{11}\det\left(\ell_{22}\det\left(\ell_{33}\det\left(L_{44}\right)\right)\right) \\ &= \ell_{11}\det\left(\ell_{22}\det\left(\ell_{33}\det\left(\dots\ell_{n-2n-2}\det\left(L_{n-1n-1}\right)\right)\right)\right) \\ &= \ell_{11}\det\left(\ell_{22}\det\left(\ell_{33}\det\left(\dots\ell_{n-2n-2}\begin{vmatrix} \ell_{n-1n-1} & 0 \\ \ell_{nn-1} & \ell_{nn} \end{vmatrix}\right)\right)\right) \end{aligned} \tag{39}$$

By expanding the terms step by step, one can further deduce that

$$\det\left(L\right) = \ell_{11}\ell_{22}\dots\ell_{n-1n-1}\ell_{nn} \tag{40}$$

The statement is hence proved. □

(c) *Prove that* $\det(A^{-1}) = \dfrac{1}{\det(A)}$.

Solution. Given the fact that

$$AA^{-1} = I \tag{41}$$

Using the property of determinants one have

$$\begin{aligned} \det(A)\det(A^{-1}) &= \det(AA^{-1}) \\ &= \det(I) = 1 \end{aligned} \tag{42}$$

Hence, it can be easily seen that

$$\det(A^{-1}) = \frac{1}{\det(A)} \tag{43}$$

The statement is hence proved. □

(d) *Let*

$$A = \begin{bmatrix} 1 & -1 & 0 \\ 3 & 1 & 2 \\ 2 & 4 & 3 \end{bmatrix}.$$

Compute $\det(A)$ *and determine the number of solutions to* $Ax = 0$.

Solution. Calculating the determinant one has

$$\begin{aligned} \det(A) &= 1\begin{vmatrix} 1 & 2 \\ 4 & 3 \end{vmatrix} - (-1)\begin{vmatrix} 3 & 2 \\ 2 & 3 \end{vmatrix} + 0 \\ &= (3-8) + (9-4) \\ &= -5 + 5 \\ &= 0 \end{aligned} \tag{44}$$

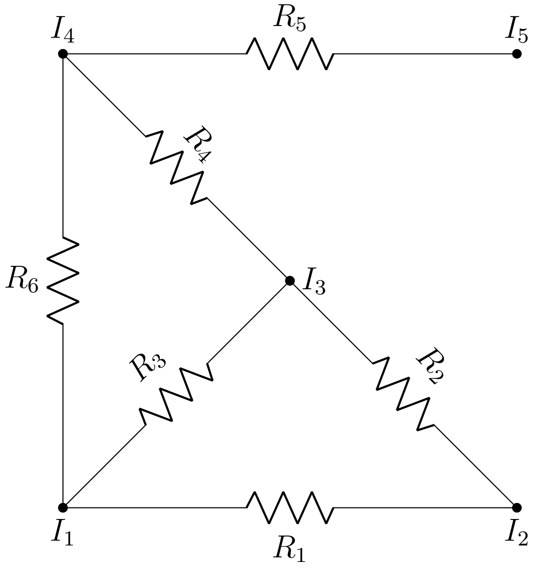
Since the determinant is zero, one deduces that there will be an infinite number of solutions for $A\vec{x} = 0$. □

Problem 5. *(Ohm's Law.)*

Suppose we have two nodes connected by a wire with resistance R, measured in ohms. Ohm's law states that the current I_{ij}, measured in amperes, traveling from node i to node j is

$$I_{ij} = \frac{V_i - V_j}{R}$$

with V_i and V_j denoting the potential at nodes i and j, both measured in volts. Notice that current is a signed quantity, which means it can be either positive or negative, so it indicates the direction of flow. Consider the following circuit.



Suppose we know the resistance in each of the 6 wires is $R = 1$ and that the potential at node i is some constant V_i.

(a) *Let I_i denote the current at node i. Recalling Kirchoff's principle, which states that I_i is the sum of all currents entering or leaving node i, express each I_i as a linear combination of the voltages V_j.*

Solution. One can first write out the matrix I:

$$I = \begin{bmatrix} 0 & \frac{V_1-V_2}{R_1} & \frac{V_1-V_3}{R_3} & \frac{V_1-V_4}{R_6} & 0 \\ \frac{V_2-V_1}{R_1} & 0 & \frac{V_2-V_3}{R_2} & 0 & 0 \\ \frac{V_3-V_1}{R_3} & \frac{V_3-V_2}{R_3} & 0 & \frac{V_3-V_4}{R_4} & 0 \\ \frac{V_4-V_1}{R_6} & 0 & \frac{V_4-V_3}{R_4} & 0 & \frac{V_4-V_5}{R_5} \\ 0 & 0 & 0 & \frac{V_5-V_4}{R_5} & 0 \end{bmatrix} \tag{45}$$

By substituting $R = 1$ one can then write out the forms for each row of I:

$$\begin{aligned} I_1 &= 3V_1 - V_2 - V_3 - V_4 \\ I_2 &= 2V_2 - V_1 - V_3 \\ I_3 &= 3V_3 - V_2 - V_1 - V_4 \\ I_4 &= 3V_4 - V_3 - V_1 - V_5 \\ I_5 &= V_5 - V_4 \end{aligned} \tag{46}$$

□

(b) *Set up a linear system from (a) as a single matrix equation. That is, find a matrix A such that $\mathbf{I} = A\mathbf{V}$.*

Solution. From (a), one can write out the coefficient matrix A:

$$A = \begin{bmatrix} 3 & -1 & -1 & -1 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 \\ -1 & 0 & -1 & 3 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix} \tag{47}$$

By multiplying the vector $\vec{V}$ one can verify that this coefficient matrix satisfy the condtion

$$\vec{I} = A\vec{V} \tag{48}$$

□

(c) *Show that the matrix you found in (b) is singular by computing its determinant. Then find a basis for its nullspace. You may use MATLAB.*

Solution. To show this matrix is singular, one can calculate the determinant of A in MATLAB:

```
>> A = [3 -1 -1 -1 0; -1 2 -1 0 0; -1 -1 3 -1 0; -1 0 -1 3 -1; 0 0 0 -1 1];
>> det(A)
ans =

     0
```

To find a basis for its nullspace, one can also calculate the nullspace using MATLAB:

```
null(A)

ans =

    0.4472
    0.4472
    0.4472
    0.4472
    0.4472
```

We then know that the nullspace is non-zero, and the basis can be written as the form

$$\mathcal{B}_{null} = \left\{ \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \right\} \tag{49}$$

.

□

(d) *Finally, describe the set of current vectors* **I** *for which the linear system you wrote down in (b) is consistent. That is, find a condition on* **I** *for which your linear system always has a solution.*

Solution. By calculating the rank (`rank(A)`) we know the rank of A is 4. Hence, the row vectors of A are linearly dependent. From (c) we already know the basis of the nullspace as a "ones-vector". Hence, we know that the linear combination of the row vectors of A with a coefficient of 1 should be a zero vector, i.e. $\vec{A}_1+\vec{A}_2+\vec{A}_3+\vec{A}_4+\vec{A}_5 = \vec{0}$.[1]

Based on this, in order for $\vec{I} = A\vec{V}$ to always have a solution, the vector $\vec{I}$ also needs to satisfy the linear combination relationship:

$$I_1 + I_2 + I_3 + I_4 + I_5 = 0 \tag{50}$$

□

[1] $\vec{A}_i$ denotes the row vectors of A

ME300A HW#4
Hanfeng Zhai (hzhai@stanford.edu) December 1, 2023

Problem 1. *One of your friends has invented a new iterative scheme for solving the system of equations $A\vec{x} = \vec{b}$ for real $n \times n$ matrices A. The scheme is given by*

$$\vec{x}^{(k+1)} = (I + \beta A)\vec{x}^{(k)} - \beta\vec{b}, \quad \textit{with } \beta > 0. \tag{1}$$

(a) *Show that if this scheme converges, it converges to the desired solution of the system of equations. In other words, your friend seems to be on to something.*

Solution. One can rewrite the update scheme:

$$\begin{aligned} \vec{x}^{(k+1)} - \vec{x}^{(k)} &= (1 + \beta A)\vec{x}^{(k)} - \beta\vec{b} - \vec{x}^{(k)} \\ &= \beta A\vec{x}^{(k)} - \beta\vec{b} \\ &= \beta(A\vec{x}^{(k)} - \vec{b}) \end{aligned} \tag{2}$$

Here, we assume that when $k \to \infty$, the system converges to the correct solution. Since the iteration scheme is proposed to solve the linear system $A\vec{x} = \vec{b}$, one can write out

$$\lim_{k\to\infty} (A\vec{x}^{(k)} - \vec{b}) = 0 \tag{3}$$

Therefore, one knows

$$\begin{aligned} \lim_{k\to\infty} \left(\vec{x}^{k+1} - \vec{x}^{(k)}\right) &= \lim_{k\to\infty} \left(\beta\left(A\vec{x}^{(k)} - \vec{b}\right)\right) \\ &= \beta \lim_{k\to\infty} \left(A\vec{x}^{(k)} - \vec{b}\right) \\ &= 0 \end{aligned} \tag{4}$$

Indicating the algorithm converges.

□

(b) *Derive an equation for the error $\vec{e}^{(k)} = \vec{x}^{(k)} - \vec{x}^*$, where $\vec{x}^*$ is the exact solution, for each iteration step k.*

Solution. We write:

$$\begin{aligned} \frac{\vec{x}^{(k+1)}}{\vec{x}^{(k)}} &= (I + \beta A) - \frac{\beta\vec{b}}{\vec{x}^{(k)}} \\ \frac{\vec{x}^{(k+1)} + \beta\vec{b}}{\vec{x}^{(k)}} &= I + \beta A \end{aligned} \tag{5}$$

We then have

$$\begin{aligned} \frac{e^{(k+1)}}{e^{(k)}} &= \frac{\vec{x}^{(k)} - \vec{x}^*}{\vec{x}^{(k-1)} - \vec{x}^*} \\ &= \frac{\vec{x}^{(k-1)} - \vec{x}^* + \beta(A\vec{x}^{(k-1)} - \vec{b})}{\vec{x}^{(k-1)} - \vec{x}^*} \\ &= 1 + \beta\frac{A\vec{x}^{(k-1)} - \vec{b}}{\vec{x}^{(k-1)} - \vec{x}^*} \end{aligned} \tag{6}$$

Since we know that $\vec{x}^*$ is the exact solution, we know $\vec{x}^* = A^{-1}\vec{b}$. We can substitute the relation back and get:

$$\begin{aligned} \frac{e^{(k+1)}}{e^{(k)}} &= 1 + \beta \frac{A\vec{x}^{(k-1)} - \vec{b}}{\vec{x}^{(k-1)} - A^{-1}\vec{b}} \\ &= 1 + \beta \frac{A\left(\vec{x}^{(k-1)} - A^{-1}\vec{b}\right)}{\vec{x}^{(k-1)} - A^{-1}\vec{b}} \\ &= 1 + \beta A \end{aligned} \tag{7}$$

Hence, we can write out the general form of $e^{(k)}$:

$$\begin{aligned} e^{(k)} &= (I + \beta A)e^{(k-1)} \\ &= (I + \beta A)^k e^{(0)} \end{aligned} \tag{8}$$

where $e^{(0)} = x^{(0)} - x^*$.

If A is not guaranteed to be non-singular (or A^{-1} is not guaranteed to exist), then the general form of the error is

$$\begin{aligned} e^{(k)} &= (I + \beta A)\vec{x}^{(k-1)} - (\vec{x}^* + \vec{b}) \\ &= (I + \beta A)^k \vec{x}^{(0)} - (\vec{x}^* + \vec{b}) \end{aligned} \tag{9}$$

which is the general form of the error. □

(c) *Does the scheme work for non–singular matrices? Explain.*

Solution. This iteration scheme does not necessarily work for all non-singular matrices.

Taking the previously derived expression for the error:

$$\begin{aligned} e^{(k)} &= e^{(k-1)} + \beta A e^{(k-1)} \\ e^{(k)} - e^{(k-1)} &= \beta A e^{(k-1)} \end{aligned} \tag{10}$$

The success of the iteration scheme for non-singular matrices depends on the choice of the parameter β and the spectral radius of $I + \beta A$.

For the scheme to converge, the spectral radius of the iteration matrix $\rho(I + \beta A)$ must be less than 1. However, here there is no guarantee that the spectral radius will be smaller than 1.

If one were to dig deeper into the convergence of this iteration scheme, one can write out the norm (one may assume an L2 norm) for the error at k^{th} from $(k-1)^{\text{th}}$ iteration:

$$\left\|\vec{e}^k\right\| / \left\|\vec{e}^{k-1}\right\| = \|(I + \beta A)\| \tag{11}$$

where its spectral radius writes:

$$\rho\left(I + \beta A\right) = \rho\left(\begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ & & & \vdots & \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix} + \beta \begin{bmatrix} a_{11} & a_{12} & & \dots & a_{1n} \\ a_{21} & a_{22} & & \dots & a_{2n} \\ & & & \vdots & \\ a_{n1} & a_{n2} & & \dots & a_{nn} \end{bmatrix}\right) \tag{12}$$

□

Problem 2. *Many modern machine learning models rely on Deep Neural Networks (DNNs) to fit complex functions defined by real-world data sets. In practice, thousands of weights parametrize a DNN and we "train" a model by finding "optimal" values for the model parameters. The "optimal" parameter values are determined by minimizing the model error as measured by a given loss function.*

The following example will motivate the usefulness of neural networks in data fitting.

Consider the following data set.

	$x^{(1)}$	$x^{(2)}$	$x^{(3)}$	$x^{(4)}$
x_1	0	0	1	1
x_2	0	1	0	1
x_3	1	1	1	1
y^T	0	1	1	0

The data in Table 2 represents a sample of $m = 4$ input-output pairs $(x^{(k)}, y_k)$, $k = 1, \ldots, m$, corresponding to the function $f : \{0,1\}^3 \to \{0,1\}$ defined by

$$f(x) = \begin{cases} 0, & \text{if } x_1 + x_2 + x_3 \text{ is odd,} \\ 1, & \text{if } x_1 + x_2 + x_3 \text{ is even.} \end{cases}$$

Each $x^{(k)}$ belongs to $\mathbb{R}^3$ and each y_k is a scalar. The domain $\{0,1\}^3$ of f is the subset of vectors in $\mathbb{R}^3$ such that each component is either 0 or 1.

We would like to "learn" f using the sample data in Table 2. In other words, we aim to fit a model g, parametrized by some weights w, to the data in Table 2 by minimizing a given loss function using gradient descent. Once we have trained our model g, we hope to use optimal parameters $\vec{w}^*$ to mimic f, so that

$$g(x; \vec{w}^*) \approx f(x)$$

for $x \in \{0,1\}^3$.

(a) We begin by fitting a linear model. Let $g : \mathbb{R}^3 \to \mathbb{R}$ denote the function defined by

$$g(x; w) = w_1 x_1 + w_2 x_2 + w_3 x_3 = w^\mathsf{T} x.$$

In this case, we package the model weights w_1, w_2, w_3 in a single vector

$$w = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}.$$

We seek parameter values w_1, w_2, w_3 minimizing the mean squared error $J(w)$, defined by

$$J(w) = \frac{1}{2m} \sum_{k=1}^{m} \left(g(x^{(k)}; w) - y_k \right)^2.$$

(i) Compute $\nabla_w J(w)$, the gradient of J with respect to w.

Solution. One may begin with expanding all the terms in $J(w)$:

$$\begin{aligned} J(w) = \frac{1}{8}\Bigg[& \left(w_1 x_1^{(1)} + w_2 x_2^{(1)} + w_3 x_3^{(1)} - y_1\right)^2 \\ & + \left(w_1 x_1^{(2)} + w_2 x_2^{(2)} + w_3 x_3^{(2)} - y_2\right)^2 \\ & + \left(w_1 x_1^{(3)} + w_2 x_2^{(3)} + w_3 x_3^{(3)} - y_3\right)^2 \\ & + \left(w_1 x_1^{(4)} + w_2 x_2^{(4)} + w_3 x_3^{(4)} - y_4\right)^2 \Bigg] \end{aligned} \tag{13}$$

To compute the gradient of $J(w)$, one computes the partial derivatives of J w.r.t. w_1, w_2 and w_3, respectively:

$$\begin{aligned} \frac{\partial J}{\partial w_1} &= \frac{1}{m}\sum_{k=1}^{m}\left(g(x^{(k)};w) - y_k\right)\frac{\partial g(x^{(k)};w)}{\partial w_1} \\ \frac{\partial J}{\partial w_2} &= \frac{1}{m}\sum_{k=1}^{m}\left(g(x^{(k)};w) - y_k\right)\frac{\partial g(x^{(k)};w)}{\partial w_2} \\ \frac{\partial J}{\partial w_3} &= \frac{1}{m}\sum_{k=1}^{m}\left(g(x^{(k)};w) - y_k\right)\frac{\partial g(x^{(k)};w)}{\partial w_3} \end{aligned} \tag{14}$$

By further derivation:

$$\begin{aligned} \frac{\partial J}{\partial w_1} &= \frac{1}{m}\sum_{k=1}^{m} x_1^{(k)}\left(w_1 x_1^{(k)} + w_2 x_2^{(k)} + w_3 x_3^{(k)} - y_k\right) \\ \frac{\partial J}{\partial w_2} &= \frac{1}{m}\sum_{k=1}^{m} x_2^{(k)}\left(w_1 x_1^{(k)} + w_2 x_2^{(k)} + w_3 x_3^{(k)} - y_k\right) \\ \frac{\partial J}{\partial w_3} &= \frac{1}{m}\sum_{k=1}^{m} x_3^{(k)}\left(w_1 x_1^{(k)} + w_2 x_2^{(k)} + w_3 x_3^{(k)} - y_k\right) \end{aligned} \tag{15}$$

By reorganizing the terms we get the general formula of the gradient based on the given form of $J(w)$:

$$\Delta_w J = \begin{bmatrix} \frac{1}{m}\sum_{k=1}^{m} x_1^{(k)}\left(w_1 x_1^{(k)} + w_2 x_2^{(k)} + w_3 x_3^{(k)} - y_k\right) \\ \frac{1}{m}\sum_{k=1}^{m} x_2^{(k)}\left(w_1 x_1^{(k)} + w_2 x_2^{(k)} + w_3 x_3^{(k)} - y_k\right) \\ \frac{1}{m}\sum_{k=1}^{m} x_3^{(k)}\left(w_1 x_1^{(k)} + w_2 x_2^{(k)} + w_3 x_3^{(k)} - y_k\right) \end{bmatrix} \tag{16}$$

for the given input-output pairs (x^k, y_k). □

(ii) Use the data points $(x^{(k)}, y_k)$, $k = 1, 2, 3, 4$, given in Table 2 and implement the gradient descent method to find w minimizing the mean squared error $J(w)$.

Compute and report the optimal $\vec{w}^*$.

Use a constant learning rate (step size) of 0.1 *and perform at least* 1500 *iterations of gradient descent. Initialize each model parameter as a uniformly distributed random number in the interval* $(0, 1)$*. In MATLAB, you may initialize* w *using* $w =$ *`rand(3, 1)`.*

Include any relevant code.

Solution. Given the instructions, we use gradient descent with a constant learning rate of 0.1 for 1500 iterations.

The relevant codes are attached herein:

```
clear; clc
%%
w = [w1; w2; w3];
x1_data = [0 0 1 1]';
x2_data = [0 1 0 1]';
x3_data = [1 1 1 1]';
y_data = [0 1 1 0]';
X = [x1_data,x2_data,x3_data];

J = .5*mse(X*w, y_data);
dJ = [diff(J,w1);diff(J,w2);diff(J,w3)];
alpha = 0.1;

%% ii
i=1;
w = rand(3, 1);
while i<=1500
    dJw = subs(dJ,{w1,w2,w3},{w(1),w(2),w(3)});
    dJw = round(dJw*1000)/1000;
    w = w-alpha*dJw;
    i = i+1;
end
```

We obtain $\vec{w}^* = \begin{bmatrix} 0.0030 \\ 0.0030 \\ 0.4965 \end{bmatrix}$. If we were to apply the solution scheme for 5000 iterations, we get $\vec{w}^* = \begin{bmatrix} 0.0028 \\ 0.0028 \\ 0.4968 \end{bmatrix}$, which is very similar to what we get for 1500 iterations. □

(iii) In this case, since g *is a linear model, we may solve for the optimal weights analytically.*

Obtain the optimal parameter values by solving the normal equations to verify the correctness of your gradient descent implementation. Include any relevant code.

Solution. Recall the normal equation for the least square method for a linear system $X\vec{w} = \vec{y}$:

$$\vec{w} = \left(X^{\mathsf{T}} X\right)^{-1} X^{\mathsf{T}} \vec{y} \tag{17}$$

One can solve it analytically by expanding the terms:

$$\begin{aligned}
\vec{w} &= \left(\begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} \right)^{-1} \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 & -\frac{1}{2} \\ 0 & 1 & -\frac{1}{2} \\ -\frac{1}{2} & -\frac{1}{2} & \frac{3}{4} \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \\
&= \begin{bmatrix} 0 \\ 0 \\ \frac{1}{2} \end{bmatrix}
\end{aligned} \tag{18}$$

One can also generate the following MATLAB codes to compute the analytical solution for $\vec{w}^*$:

```
%% iii
X = [0 0 1; 0 1 1; 1 0 1; 1 1 1];
y = [0;1;1;0];
w_anal = inv(X'*X)*X'*y;
```

and obtain the corresponding solution $\vec{w}^* = \begin{bmatrix} 0 \\ 0 \\ 0.5 \end{bmatrix}$. One then deduces that the numerical solution obtained from gradient descent is accurate as it is close to the analytical solution. □

(iv) *Use the optimal parameters $\vec{w}^*$ obtained in (ii) to evaluate $g(x^{(1)}; \vec{w}^*)$.*

Since we are fitting data sampled from the function f, we hope to obtain $f(x^{(1)}) = 0$. However, you will find that our linear model is inadequate.

Solution. Using the numerical linear model with the approximation results of 1500 iterations, we compute $f(x^{(1)})$:

$$\begin{aligned}
f\left(x^{(1)}\right) &\approx g\left(x^{(1)}; w^*\right) \\
&= w_1 \cdot 0 + w_2 \cdot 0 + w_3 \cdot 1 \\
&= 0.4965
\end{aligned} \tag{19}$$

Since we know that in the real data $f\left(x^{(1)}\right) = 0$. We therefore know that the fitted data is inaccurate, and hence our linear model is inadequate. □

(b) *We now consider a non-linear model g. We begin with a few definitions.*

Let $\sigma : \mathbb{R} \to \mathbb{R}$ denote the sigmoid function, defined by

$$\sigma(x) = \frac{1}{1 + e^{-x}}.$$

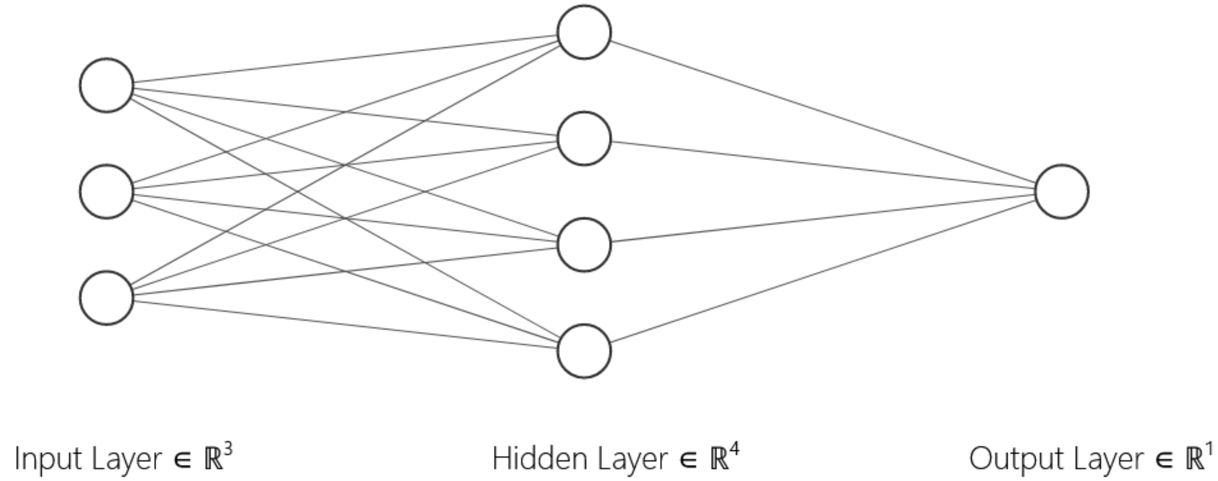
Let h denote the so-called number of hidden units and let $\nu : \mathbb{R}^h \to \mathbb{R}^h$ denote the vectorization of σ, defined by

$$\nu(z) = \begin{bmatrix} \sigma(z_1) \\ \sigma(z_2) \\ \vdots \\ \sigma(z_h) \end{bmatrix}.$$

Let $g : \mathbb{R}^3 \to \mathbb{R}$ denote the fully connected two-layer feed-forward neural network defined by

$$g(x; \alpha, W) = \sigma\left(\alpha^\mathsf{T} \nu(Wx)\right). \tag{20}$$

This model is parametrized by the weights α_j, for $j = 1, \ldots, h$, and w_{ij}, for $i = 1, \ldots, h$ and $j = 1, 2, 3$. Also known as a Multi-Layer Perceptron (MLP) head with a single hidden layer, the network defined by g is illustrated in the Figure in case $h = 4$.


Input Layer ∈ $\mathbb{R}^3$
Hidden Layer ∈ $\mathbb{R}^4$
Output Layer ∈ $\mathbb{R}^1$

In this case, it will be convenient to package the model parameters in a $1 \times h$ row vector

$$\alpha^\mathsf{T} = \begin{bmatrix} \alpha_1 & \alpha_2 & \cdots & \alpha_h \end{bmatrix}$$

and an $h \times 3$ matrix

$$W = \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ \vdots & \vdots & \vdots \\ w_{h1} & w_{h2} & w_{h3} \end{bmatrix}.$$

We will fit g to the data in Table 2 using the loss function $L(\alpha^\mathsf{T}, W)$ defined by

$$L(\alpha^\mathsf{T}, W) = \sum_{k=1}^{m} \left(g(x^{(k)}; \alpha^\mathsf{T}, W) - y_k\right)^2.$$

We will use gradient descent to find optimal parameter values. In order to implement the gradient descent method, it will be convenient to package the partial derivatives of the loss function with respect to our model parameters into the two gradients

$$\nabla_\alpha L(\alpha^\intercal, W) = \begin{bmatrix} \frac{\partial L}{\partial \alpha_1} & \cdots & \frac{\partial L}{\partial \alpha_h} \end{bmatrix}, \text{ and}$$

$$\nabla_W L(\alpha^\intercal, W) = \begin{bmatrix} \frac{\partial L}{\partial w_{11}} & \frac{\partial L}{\partial w_{12}} & \frac{\partial L}{\partial w_{13}} \\ \vdots & \vdots & \vdots \\ \frac{\partial L}{\partial w_{h1}} & \frac{\partial L}{\partial w_{h2}} & \frac{\partial L}{\partial w_{h3}} \end{bmatrix}.$$

Given these gradients, we will update our model parameters $\alpha^{(n)}$ and $W^{(n)}$ at the nth step of the gradient descent algorithm using

$$\begin{aligned} (\alpha^{(n+1)})^\intercal &\leftarrow (\alpha^{(n)})^\intercal - \nabla_\alpha L\left((\alpha^{(n)})^\intercal, W^{(n)}\right), \\ W^{(n+1)} &\leftarrow W^{(n)} - \nabla_W L\left((\alpha^{(n)})^\intercal, W^{(n)}\right). \end{aligned} \tag{21}$$

We now turn to computing these gradients.

(i) Begin by showing that $\sigma'(x) = \sigma(x)(1 - \sigma(x))$.

Solution. To show the given expression, we begin with expanding the terms in $\sigma'(x)$ (LHS):

$$\begin{aligned} \sigma'(x) &= \frac{d}{dx}\frac{1}{1+e^{-x}} \\ &= \frac{e^{-x}}{(1+e^{-x})^2} \\ &= \frac{1}{1+e^{-x}} \cdot \frac{e^{-x}}{1+e^{-x}} \\ &= \frac{1}{1+e^{-x}} \cdot \frac{1+e^{-x}-1}{1+e^{-x}} \\ &= \frac{1}{1+e^{-x}}\left(1 - \frac{1}{1+e^{-x}}\right) \end{aligned} \tag{22}$$

which can be rearranged as the original form of the RHS:

$$\sigma'(x) = \sigma(x)\left(1 - \sigma(x)\right) \tag{23}$$

The statement is hence proved. □

(ii) Next, let y denote a given vector in $\mathbb{R}^h$ and compute

$$\frac{\partial}{\partial \alpha_j}\left[\sigma(\alpha^\intercal y)\right] = \frac{\partial}{\partial \alpha_j}\left[\sigma(\alpha_1 y_1 + \cdots + \alpha_h y_h)\right]$$

for each $j = 1, \ldots, h$.

Let $\phi(x;W):\mathbb{R}^3\to\mathbb{R}^h$ denote the output of the first layer of our neural network, defined by the composition

$$\phi(x;W)=\nu(Wx).$$

We will use the shorthand $\phi^{(k)}=\phi(x^{(k)};W)$, and as usual we denote the jth component of the $h\times 1$ vector $\phi^{(k)}$ by $\phi_j^{(k)}$.

Solution. We may begin by expanding the general form of the LHS:

$$\begin{aligned}
\frac{\partial}{\partial\alpha_j}\left[\sigma(\alpha^\mathsf{T}y)\right] &= \frac{\partial}{\partial\alpha_j}\left[\sigma\left(\begin{bmatrix}\alpha_1 & \alpha_2 & \dots & \alpha_h\end{bmatrix}\begin{bmatrix}y_1\\ y_2\\ \vdots\\ y_h\end{bmatrix}\right)\right]\\
&= \frac{\partial}{\partial\alpha_j}\left[\frac{1}{1+e^{-(\alpha_1y_1+\alpha_2y_2+\ldots+\alpha_hy_h)}}\right]\\
&= \frac{\partial}{\partial\alpha_j}\left[\frac{1}{1+e^{-\sum_{i=1}^h\alpha_iy_i}}\right]\\
&= \begin{bmatrix} y_1\frac{e^{-\sum_{i=1}^h\alpha_iy_i}}{\left(1+e^{-\sum_{i=1}^h\alpha_iy_i}\right)^2}\\ y_2\frac{e^{-\sum_{i=1}^h\alpha_iy_i}}{\left(1+e^{-\sum_{i=1}^h\alpha_iy_i}\right)^2}\\ \vdots\\ y_h\frac{e^{-\sum_{i=1}^h\alpha_iy_i}}{\left(1+e^{-\sum_{i=1}^h\alpha_iy_i}\right)^2}\end{bmatrix}
\end{aligned}\tag{24}$$

The general form can be written as

$$\frac{\partial}{\partial\alpha_j}\left[\sigma(\alpha^\mathsf{T}y)\right]=y_j\frac{e^{-\sum_{j=1}^h\alpha_jy_j}}{\left(1+e^{-\sum_{j=1}^h\alpha_iy_i}\right)^2}\tag{25}$$

Since we know from the previous proof that

$$\frac{e^{-\sum_{j=1}^h\alpha_jy_j}}{\left(1+e^{-\sum_{j=1}^h\alpha_iy_i}\right)^2}=\sigma'\left(\sum_{j=1}^h\alpha_iy_i\right)\tag{26}$$

Hence, one can write out the general form of the partial derivative:

$$\frac{\partial}{\partial\alpha_j}\left[\sigma(\alpha^\mathsf{T}y)\right]=y_j\sigma'\left(\sum_{j=1}^h\alpha_iy_i\right)\tag{27}$$

It can be further expanded as

$$\frac{\partial}{\partial\alpha_j}\left[\sigma(\alpha^\mathsf{T}y)\right]=\sigma\left(\alpha^\mathsf{T}y\right)\left(1-\sigma\left(\alpha^\mathsf{T}y\right)\right)y_j\tag{28}$$

□

(iii) Use the chain rule to show that

$$\frac{\partial L}{\partial \alpha_j} = 2\sum_{k=1}^{m}\left(g(x^{(k)};\alpha^\mathsf{T}, W) - y_k\right)\sigma'\left(\alpha^\mathsf{T}\phi^{(k)}\right)\phi_j^{(k)}.$$

Solution. We may begin by writing out the general form of L:

$$\begin{aligned} L &= \sum_{k=1}^{m}\left(g(x^{(k)};\alpha^\mathsf{T}, W) - y_k\right)^2 \\ &= \sum_{k=1}^{m}\left[\sigma\left(\alpha_i\sigma\left(Wx_i^{(k)}\right)\right) - y_k\right]^2 \end{aligned} \tag{29}$$

Using the chain rule, we can rewrite the loss function as

$$\frac{\partial L}{\partial \alpha_j} = \frac{\partial L}{\partial g}\frac{\partial g}{\partial \alpha_j} \tag{30}$$

We may get some intuition by expanding the general form of g:

$$g = \frac{1}{1 + e^{-\alpha^\mathsf{T}\begin{bmatrix}\sigma(Wx_1^{(k)}) \\ \sigma(Wx_2^{(k)}) \\ \vdots \\ \sigma(Wx_h^{(k)})\end{bmatrix}}} \tag{31}$$

Or simply

$$g = \frac{1}{1 + e^{-\alpha^\mathsf{T}\left(Wx_j^{(k)}\right)}} \tag{32}$$

By computing the partial derivative of g one gets:

$$\begin{aligned} \frac{\partial g}{\partial \alpha_j} &= \frac{-(-1)e^{-\alpha^\mathsf{T}\sigma\left(Wx_j^{(k)}\right)}}{(1 + e^{-\alpha^\mathsf{T}\sigma\left(Wx_j^{(k)}\right)})^2}\sigma\left(Wx_j^{(k)}\right) \\ &= \frac{\sigma\left(Wx_j^{(k)}\right)e^{-\alpha^\mathsf{T}\sigma\left(Wx_j^{(k)}\right)}}{\left(1 + e^{-\alpha^\mathsf{T}\sigma\left(Wx_j^{(k)}\right)}\right)^2} \\ &= \sigma\left(Wx_j^{(k)}\right)\sigma'(-\alpha^\mathsf{T}\sigma\left(Wx_j^{(k)}\right) \\ &= \sigma'(\alpha^\mathsf{T}\phi^{(k)})\phi_j^{(k)} \end{aligned} \tag{33}$$

One can also expand the form of $\frac{\partial L}{\partial g}$:

$$\begin{aligned}\frac{\partial L}{\partial g} &= \frac{\partial\left(\sum_{k=1}^{m}(g_k - y_k)^2\right)}{\partial g_k}\\ &= 2\sum_{k=1}^{m}(g_k - y_k)\\ &= 2\sum_{k=1}^{m}\left(g(x^{(k)};\alpha^{\mathsf{T}},W) - y_k\right)\end{aligned}\tag{34}$$

Applying the chain rule and concatenate the two terms one has:

$$\begin{aligned}\frac{\partial L}{\partial \alpha_j} &= \frac{\partial L}{\partial g}\frac{\partial g}{\partial \alpha_j}\\ &= 2\sum_{k=1}^{m}\left(g(x^{(k)};\alpha^{\mathsf{T}},W) - y_k\right)\sigma'\left(\alpha^{\mathsf{T}}\phi^{(k)}\right)\phi_j^{(k)}\end{aligned}\tag{35}$$

The statement is hence proved. □

Keeping (iii) in mind, notice that the $1\times h$ *gradient* $\nabla_\alpha L(\alpha^{\mathsf{T}},W)$ *can be written as the vector-matrix product*

$$\nabla_\alpha L(\alpha^{\mathsf{T}},W) = 2\left((\vec{g} - y^{\mathsf{T}})\star v^{\mathsf{T}}\right)\Phi^{\mathsf{T}},$$

where $\vec{g}$ *denotes the* $1\times m$ *row vector*

$$\vec{g} = \begin{bmatrix} g(x^{(1)};\alpha^{\mathsf{T}},W) & \cdots & g(x^{(m)};\alpha^{\mathsf{T}},W)\end{bmatrix},$$

v^{T} *is an appropriately defined* $1\times m$ *row vector, and* Φ *denotes the* $h\times m$ *matrix*

$$\begin{aligned}\Phi &= [\phi(x^{(1)};W),\ldots,\phi(x^{(m)};W)]\\ &= [\phi^{(1)},\ldots,\phi^{(m)}].\end{aligned}$$

Here $\star$ *denotes the element-wise vector product, so that for any* $1\times h$ *row vectors* a *and* b*, the product* $a\star b$ *is again a* $1\times h$ *row vector with*

$$(a\star b)_j = a_j b_j.$$

(iv) Next, use the chain rule to compute $\frac{\partial L}{\partial w_{ij}}$.

The calculation in part (ii) will serve as a motivating blueprint.

Solution. Using the chain rule, we can write

$$\frac{\partial L}{\partial w_{ij}} = \frac{\partial L}{\partial g}\frac{\partial g}{\partial \phi}\frac{\partial \phi}{\partial w_{ij}}\tag{36}$$

We first expand the last term $\frac{\partial \phi}{\partial w_{ij}}$:

$$\frac{\partial \phi}{\partial w_{ij}} = \frac{\partial \begin{bmatrix} \sigma\left(\begin{bmatrix} w_{11} & \dots \\ \dots & w_{ij} \end{bmatrix} x_1^{(k)}\right) \\ \sigma\left(\begin{bmatrix} w_{11} & \dots \\ \dots & w_{ij} \end{bmatrix} x_2^{(k)}\right) \\ \vdots \\ \sigma\left(\begin{bmatrix} w_{11} & \dots \\ \dots & w_{ij} \end{bmatrix} x_h^{(k)}\right) \end{bmatrix}}{\partial w_{ij}} \tag{37}$$

Or in the general form:

$$\frac{\partial \phi}{\partial w_{ij}} = \sigma'\left(w_{ij} x_j^{(k)}\right) x_j^{(k)} \tag{38}$$

One can then deal with the second term in the chain rule expansion:

$$\begin{aligned} \frac{\partial g}{\partial \phi} &= \frac{\partial\left[\sigma\left(\alpha^{\mathsf{T}} \nu(Wx)\right)\right]}{\partial\left[\nu(Wx)\right]} \\ &= \frac{\partial\left[\frac{1}{1+e^{-\alpha^{\mathsf{T}}\nu(Wx)}}\right]}{\partial\left[\nu(Wx)\right]} \\ &= \frac{-(-\alpha^{\mathsf{T}})e^{-\alpha^{\mathsf{T}}(\nu(Wx))}}{\left(1+e^{-\alpha^{\mathsf{T}}(\nu(Wx))}\right)^2} \\ &= \alpha^{\mathsf{T}}\sigma'\left(-\alpha^{\mathsf{T}}\nu\left(Wx_j^{(k)}\right)\right) \end{aligned} \tag{39}$$

The first term in the chain rule can be obtained by recalling the previous question:

$$\frac{\partial L}{\partial g} = 2\sum_{k=1}^{m}\left(g\left(x_j^{(k)}; \alpha^{\mathsf{T}}, W\right) - y_k\right) \tag{40}$$

By concatenating the three terms back into the chain rule we have

$$\begin{aligned} \frac{\partial L}{\partial w_{ij}} &= 2\sum_{k=1}^{m}\left(g\left(x_j^{(k)}; \alpha^{\mathsf{T}}, W\right) - y_k\right)\alpha_j\sigma'\left(-\alpha^{\mathsf{T}}\nu\left(Wx_j^{(k)}\right)\right)\sigma'\left(w_{ij}x_j^{(k)}\right)x_j^{(k)} \\ &= 2\sum_{k=1}^{m}\left(g\left(x_j^{(k)}; \alpha^{\mathsf{T}}, W\right) - y_k\right)\alpha_j\sigma'\left(-\alpha^{\mathsf{T}}\phi_j^{(k)}\right)\sigma'\left(w_{ij}x_j^{(k)}\right)x_j^{(k)} \end{aligned} \tag{41}$$

Specifically for our case, with 4 hidden layers and 4 data sets with three fitting parameters, the model can be written as:

$$\frac{\partial L}{\partial w_{ij}} = \sum_{k=1}^{4} 2\left(g^{(k)} - y_k\right)\sigma'\left(\sum_{i=1}^{4}\alpha_i\sigma\left(N_i^{(k)}\right)\right)\alpha_i\sigma'\left(w_{i1}x_1^{(k)} + w_{i2}x_2^{(k)} + w_{i3}x_3^{(k)}\right)x_j^{(k)} \tag{42}$$

□

(v) *For ease of implementation, we write the* $h \times 3$ *gradient* $\nabla_W L(\alpha^\mathsf{T}, W)$ *as a matrix-matrix product.*

In particular, find $h \times m$ *matrices* S *and* P *such that*

$$\nabla_W L(\alpha^\mathsf{T}, W) = 2(S \star P)X^\mathsf{T},$$

where X *denotes the* $3 \times m$ *matrix of data points:*

$$X = \begin{bmatrix} x^{(1)} & \cdots & x^{(m)} \end{bmatrix}.$$

Here $S \star P$ *denotes the element-wise product of* S *and* P*, so that*

$$(S \star P)_{ij} = s_{ij}p_{ij}.$$

Hint: The matrix P *can be expressed as an outer product.*

Solution.

Given that $\nabla_W L(\alpha^\mathsf{T}, W)$ can be written as a matrix-matrix product $2(S \star P)X^\mathsf{T}$, where S is an $h \times m$ matrix, P is an $h \times m$ matrix, X is a $3 \times m$ matrix of the data table.

Following our previous solution, recall $\frac{\partial L}{\partial w}$:

$$\frac{\partial L}{\partial w_{ij}} = 2\sum_{k=1}^{m}\left(g\left(x_j^{(k)}; \alpha^\mathsf{T}, W\right) - y_k\right)\alpha_j \sigma'\left(-\alpha^\mathsf{T}\phi_j^{(k)}\right)\sigma'\left(w_{ij}x_j^{(k)}\right)x_j^{(k)} \tag{43}$$

From the hint, by observing the other terms one may see the “outer product”:

$$p_{ij} = \alpha_j \sum_{k=1}^{m}(g_k - y_k)\sigma'\left(Wx^{(k)}\right) \tag{44}$$

where the multiplication between $(g_k - y_k)$ and $\sigma'(Wx^{(k)})$ are per element-wised. Or one may also write out the general form for P:

$$P = \alpha^\mathsf{T}\left((\vec{g} - \vec{y})\,\sigma'(WX)\right) \tag{45}$$

where X stores all the $\vec{x}$s: $X = [\vec{x}^{(1)},\ \vec{x}^{(2)},\ \vec{x}^{(3)},\ \vec{x}^{(4)}]$.

And the form for S:

$$S = \sigma'\left(\alpha^\mathsf{T}\sigma'\left(\alpha^\mathsf{T}\phi\right)\right) \tag{46}$$

Note that this is not the only way to construct S and P. □

(vi) *Implement the gradient descent method to fit your neural network* g *to the data in Table 2.*

Use $h = 4$ *hidden units and perform at least* 1500 *iterations of gradient descent, updating your model parameters at each step as described by* (21)*. Initialize each parameter by independently drawing a uniformly distributed random number in* *the interval* $(0, 1)$*. In MATLAB, you may initialize your parameters using* `alpha` $=$ `rand(1, h)`;   `W` $=$ `rand(h, 3)`;

Report optimal values for the model parameters. Report your fitted model's output for each data point in Table 2.

Include a convergence plot graphing the total loss as a function of iteration number, and include all relevant code.

Solution.

Given the hints and previous derivations, I wrote the following codes:

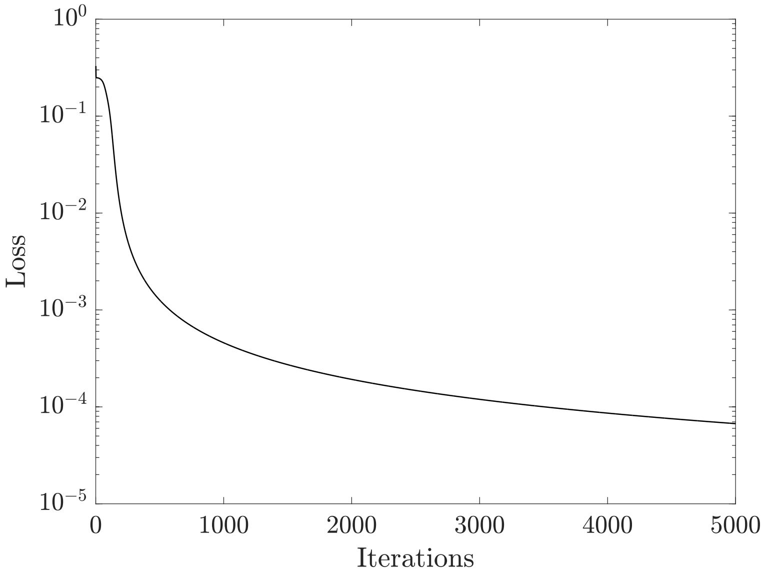
```
clear;clc

x1_data = [0 0 1 1]';
x2_data = [0 1 0 1]';
x3_data = [1 1 1 1]';
y_data = [0 1 1 0]';y = y_data;
X = [x1_data,x2_data,x3_data]';

h = 4; alpha = rand(1, h); W = rand(h, 3);
%Define helper functions
sigmoid = @(x) 1./(1 + exp(-x));
dsigmoid = @(s) s .* (1 - s);
one_layer = @(X, W) sigmoid(W * X);
nn = @(X, alpha, W) one_layer(one_layer(X, W), alpha);
phi = @(i) sigmoid(W*X(:,i));
Phi = [phi(1),phi(2),phi(3),phi(4)];
%% NN iterations

y = y_data;y = y';
fprintf("------------------------")
for iter = 1:5000
    g = nn(X, alpha, W);

    phi = one_layer(X,W);
    dL_dalpha = 2*(g - y) .* dsigmoid(g) * phi;

    S = dsigmoid(one_layer(X,W));
    P = alpha' * (g - y) .* dsigmoid(g);
    dL_dW = 2 * S.*P*X';
    fprintf("*********************");fprintf("Iteration %d",
        iter);fprintf("*********************\n")

    W = W - dL_dW;
    alpha = alpha - dL_dalpha;


    Loss(iter) = mse(nn(X, alpha, W),y);
end
y_pred = nn(X, alpha, W);
```

And after 5000 iterations, we get the output (the prediction) as

```
>> y_pred

```

```
y_pred =

    0.0069    0.9892    0.9922    0.0092
```

The convergence plot is attached in the following figure (loss was plotted in the log scale). It can be clearly observed that the loss decreases and converges to a very low value ($\sim 10^{-4}$). And the corresponding output `0.0069    0.9892    0.9922    0.0092` is very close to the given training data $y = [0 \quad 1 \quad 1 \quad 0]^{\mathsf{T}}$. Hence, the neural network worked well to converge to the desired value.



Note also this is not the only way to make the NN work. If one were to strictly stick with the hint, we may also construct two MATLAB functions "`grad1()`", "`grad2()`" as follows:

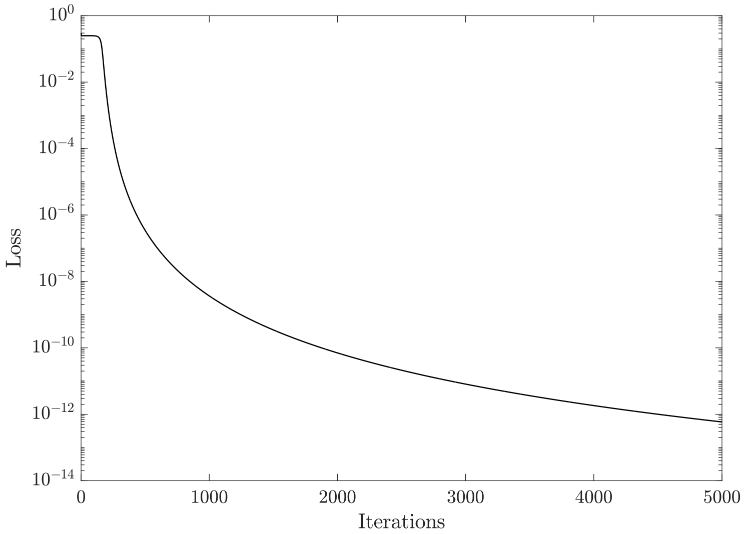
```
function dL_dW = grad1(X, y, alpha, W)
    % Helper functions
    sigmoid = @(X) 1./(1 + exp(-X));
    dsigmoid = @(s) s .* (1 - s);
    one_layer = @(X, W) sigmoid(W * X);

    nn = @(X, alpha, W) one_layer(one_layer(X, W), alpha);
    nn2 = @(alpha, Phi) one_layer(one_layer(Phi), alpha);
    g = nn(X, alpha, W);
    S = dsigmoid(one_layer(X,W));
    P = alpha' * (g - y) .* dsigmoid(g);
    dL_dW = 2 * S.*P*X';
end
```

```
function dL_dalpha = grad2(Phi, y, alpha)
    dsigmoid2 = @(s) s .* (1 - s);
    sigmoid = @(X) 1./(1 + exp(-X));
    one_layer = @(X, W) sigmoid(W * X);
    one_layer2 = @(Phi) sigmoid(Phi);
    Phi_func = @(Phi) one_layer2(Phi);
    nn2 = @(alpha, Phi) one_layer(sigmoid(Phi), alpha);

    g = nn2(alpha, Phi);
    dL_dalpha = 2*(g - y) .* dsigmoid2(g) * Phi;
end
```

Surprisingly, using this method, for 5000 iterations, I got an extremely accurate result:

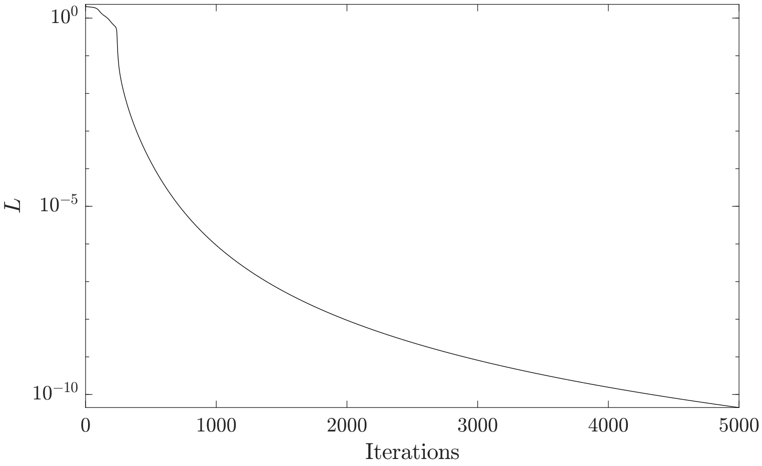
```
>> y_pred

y_pred =

    0.0000    1.0000    1.0000    0.0000
```

Using this method, the corresponding loss evolution is plotted as follows:



One observes that the loss drops to $\sim 10^{-12}$, which is extremely small. So it is found that using this “function-based” approach, the approximation accuracy has been significantly improved.

Here, the loss is plotted using the MATLAB `mse`$(\cdot)$ function, which is not directly using the loss L we defined in the instruction. One may also directly plot the corresponding convergence of loss L as follows (this is a different attempt with a different set of randomized initialization), which should show the same trend:



Both the "`mse`" loss and the defined L show the same converging trend. The corresponding reported optimal weights W and α are

```
>> W

W =

    5.0344    3.9085   -6.9991
   -2.6231    2.1599   -0.8294
    6.7636    7.3798   -3.0521
   -3.2643    3.3287    2.3026

>> alpha

alpha =

   -8.0477    5.2885    7.8228   -7.2177
```

□

(vii) Using the optimal parameter values obtained in (vi), evaluate your neural network $g(x;\alpha^{\mathsf{T}},W)$ at the point $x^{(1)}=[0,0,1]^{\mathsf{T}}$.

Report your model's prediction and compare it with your result from part (a)(iv).

Solution.

Based on the given output I printed (from Method 1) from the last sub-question, we know the corresponding evaluated y_1 is 0.0069, which is very close to 0. If one uses the prediction from my reported second method, the prediction is $y_1 = 0.0000$, which indicates that with 4-digit precision the prediction is basically the same as the training data. This result is significantly more accurate than the pure linear model prediction from (a)(iv).

Here, we may have some additional discussions for the neural network implementation. Using the function approach ("`grad1`$(\cdot)$" and "`grad2`$(\cdot)$"), the numerical accuracy is higher. If one directly computes the $\frac{\partial L}{\partial W}$ and $\frac{\partial L}{\partial \alpha}$ in the same MATLAB script, the numerical accuracy is reported lower. □

Implementation hints:

- *Built-in functions in MATLAB are vectorized, which means, for instance, that the MATLAB command* **`exp(ones(4,2))`** *applies the* **`exp`** *function to each component of the array* **`ones(4,2)`***.*
- *In MATLAB, you may perform component-wise array products and quotients by prefixing the appropriate operator with a period. For instance, the command* **`v .* w`** *computes the component-wise product of the arrays* **`v`** *and* **`w`***.*
- *The following MATLAB code might be useful. Aside from the helper functions below, all that is needed to implement gradient descent are methods* **`grad1(X, y, alpha, W)`** *and* **`grad2(Phi, y, alpha)`** *that can evaluate the relevant derivatives. Each of these can be implemented with less than 7 lines of code!*

```
%Initialize parameters
h = 4; alpha = rand(1, h); W = rand(h, 3);

%Define helper functions
sigmoid = @(x) 1./(1 + exp(-x));

dsigmoid = @(s) s .* (1 - s);

one_layer = @(X, W) sigmoid(W * X);

nn = @(X, alpha, W) one_layer(one_layer(X, W), alpha);
```

Problem 3. *(a) Compute the eigenvalues and eigenvectors of the following matrix:*

$$A = \begin{bmatrix} -1 & 3 & 1 \\ -1 & 3 & 1 \\ -3 & 3 & 3 \end{bmatrix}$$

Solution. We begin with calculating the eigenvalues:

$$\begin{aligned} \det(A - \lambda I) &= 0 \\ \begin{vmatrix} -1-\lambda & 3 & 1 \\ -1 & 3-\lambda & 1 \\ -3 & 3 & 3-\lambda \end{vmatrix} &= 0 \\ (-1-\lambda)\begin{vmatrix} 3-\lambda & 1 \\ 3 & 3-\lambda \end{vmatrix} - 3\begin{vmatrix} -1 & 1 \\ -3 & 3-\lambda \end{vmatrix} + \begin{vmatrix} -1 & 3-\lambda \\ -3 & 3 \end{vmatrix} &= 0 \end{aligned} \tag{47}$$

Expanding the equation one has

$$-(1+\lambda)(3-\lambda)^2 + 6(3-\lambda) + 3(1+\lambda) - 12 = 0 \tag{48}$$

Solving the equation one gets

$$\begin{cases} \lambda_1 = 0 \\ \lambda_2 = 2 \\ \lambda_3 = 3 \end{cases} \tag{49}$$

One can solve for the eigenvectors for the different eigenvalues respectively. For $\lambda_1 = 0$, we have

$$\begin{bmatrix} -1 & 3 & 1 \\ -1 & 3 & 1 \\ -3 & 3 & 3 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = 0 \tag{50}$$

We can then solve the systems of equations:

$$\begin{cases} 3v_2 - v_1 + v_3 = 0 \\ 3v_2 - v_1 + v_3 = 0 \\ 3v_2 - 3v_1 + 3v_3 = 0 \end{cases} \rightarrow \begin{cases} v_1 = v_3 \\ v_2 = 0 \end{cases} \tag{51}$$

One then get the first eigenvector:

$$\vec{V_1} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \tag{52}$$

Here, the normalized form of the eigenvector $\vec{V_1}$ should be

$$\vec{V_1} = \begin{bmatrix} 1/\sqrt{2} \\ 0 \\ 1/\sqrt{2} \end{bmatrix} \tag{53}$$

For $\lambda_2 = 2$, we have

$$\begin{bmatrix} -3 & 3 & 1 \\ -1 & 1 & 1 \\ -3 & 3 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = 0 \tag{54}$$

We can then solve the systems of equations:

$$\begin{cases} 3v_2 - 3v_1 + v_3 = 0 \\ v_2 - v_1 + v_3 = 0 \\ 3v_2 - 3v_1 + v_3 = 0 \end{cases} \rightarrow \begin{cases} v_1 = v_2 \\ v_3 = 0 \end{cases} \tag{55}$$

One then get the second eigenvector:

$$\vec{V}_2 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \tag{56}$$

Here, the normalized form of the eigenvector $\vec{V}_2$ should be

$$\vec{V}_2 = \begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \\ 0 \end{bmatrix} \tag{57}$$

For $\lambda_3 = 3$, we have

$$\begin{bmatrix} -4 & 3 & 1 \\ -1 & 0 & 1 \\ -3 & 3 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = 0 \tag{58}$$

We can then solve the systems of equations:

$$\begin{cases} 3v_2 - 4v_1 + v_3 = 0 \\ v_3 - v_1 = 0 \\ 3v_2 - 3v_1 = 0 \end{cases} \rightarrow \begin{cases} v_1 = v_2 \\ v_1 = v_3 \end{cases} \tag{59}$$

One then get the third eigenvector:

$$\vec{V}_3 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \tag{60}$$

Here, the normalized form of the eigenvector $\vec{V}_3$ should be

$$\vec{V}_3 = \begin{bmatrix} 1/\sqrt{3} \\ 1/\sqrt{3} \\ 1/\sqrt{3} \end{bmatrix} \tag{61}$$

The three eigenvalues λ_1, λ_2, λ_3, and three eigenvectors $\vec{V}_1$, $\vec{V}_2$, $\vec{V}_3$ are then obtained.

We may also represent them in the form of a spanning set, denoted as $\mathbf{V}$:

$$\mathbf{V} = \left\{ \begin{bmatrix} 1/\sqrt{2} \\ 0 \\ 1/\sqrt{2} \end{bmatrix}, \begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \\ 0 \end{bmatrix}, \begin{bmatrix} 1/\sqrt{3} \\ 1/\sqrt{3} \\ 1/\sqrt{3} \end{bmatrix} \right\} \tag{62}$$

□

(b) *Prove that if a symmetric matrix A has n distinct eigenvalues, then the corresponding eigenvectors are orthogonal to each other.*

Solution. Since we know that A is symmetric, and A has n distinct eigenvalues, it is then known that one can apply the canonical decomposition for A[1]:

$$A = Y\Lambda Y^{-1} \tag{63}$$

where Λ stores all the eigenvalues. We then know the matrix Y stores all the vectors. Since it is known that by definition for the canonical decomposition, the columns in Y are orthogonal. Hence the statement is proven.

One may also prove this statement without thinking about the canonical decomposition. Let's denote the symmetric matrix A with distinct eigenvalues as A and its corresponding eigenvectors as $v_1, v_2, \dots, v_n$ corresponding to eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_n$.

By definition, the eigenvalues and eigenvectors for A are given by:

$$A\vec{v}_i = \lambda_i \vec{v}_i \tag{64}$$

Now, let's consider two distinct eigenvectors $\vec{v}_i$ and $\vec{v}_j$ corresponding to eigenvalues λ_i and λ_j where $i \neq j$. We want to prove that $\vec{v}_i$ and $\vec{v}_j$ are orthogonal. In other words, we want to show that $\vec{v}_i^{\mathsf{T}} \vec{v}_j = 0$.

From the definition in Equation (64), we know that

$$(A - \lambda_i)\vec{v}_i = 0 \tag{65}$$

We can multiply Equation (65) by $\vec{v}_j$:

$$\vec{v}_j^{\mathsf{T}} A \vec{v}_i - \vec{v}_j^{\mathsf{T}} \lambda_i \vec{v}_i = 0 \tag{66}$$

Since we know that A^{T} is a symmetric matrix, we know:

$$\begin{aligned} \vec{v}_j^{\mathsf{T}} A^{\mathsf{T}} \vec{v}_i - \vec{v}_j^{\mathsf{T}} \lambda_i \vec{v}_i &= 0 \\ (A\vec{v}_j)^{\mathsf{T}} \vec{v}_i - \lambda_i \vec{v}_j^{\mathsf{T}} \vec{v}_i &= 0 \end{aligned} \tag{67}$$

Since we also know that (by definition) $A\vec{v}_j = \lambda_j \vec{v}_j$, Equation (67) can be further written as

$$\begin{aligned} \lambda_j \vec{v}_j^{\mathsf{T}} \vec{v}_i - \lambda_i \vec{v}_j^{\mathsf{T}} \vec{v}_i &= 0 \\ (\lambda_j - \lambda_i) \vec{v}_j^{\mathsf{T}} \vec{v}_i &= 0 \end{aligned} \tag{68}$$

[1] or in other words, the canonical decomposition exists

Since we already assumed that A has n distinct eigenvalues, we know that $\lambda_j \neq \lambda_i$, or $(\lambda_j - \lambda_i) \neq 0$. Hence, the only way to establish Equation (68) is

$$\vec{v}_j^{\mathsf{T}} \vec{v}_i = 0 \tag{69}$$

Hence, in this sense, we also proved that the eigenvectors of A have to be orthogonal to each other. □

(c) *Suppose that P is any invertible $n \times n$ matrix. Show that A and $P^{-1}AP$ have the same eigenvalues.*

Solution. Taking the previous assumption that A is symmetric and assume A has canonical decomposition: $A = Y\Lambda Y^{-1}$. We may define that $B = P^{-1}AP$. One can then expand B in terms of the canonical decomposition of A:

$$B = P^{-1}Y\Lambda Y^{-1}P \tag{70}$$

where Λ stores all the eigenvalues of A. One can further write this relation as

$$B = \left(P^{-1}Y\right)\Lambda\left(P^{-1}Y\right)^{-1} \tag{71}$$

where we may define $X = P^{-1}Y$, such that $B = X\Lambda X^{-1}$.

Since vectors in Y are A's eigenvectors, we know

$$(A - \lambda)\vec{y}_i = 0, \quad \vec{y}_i \in Y \tag{72}$$

or further:

$$(A - \Lambda)Y = \vec{0} \tag{73}$$

Since λ is a diagonal matrix, we know

$$\Lambda Y = Y\Lambda \tag{74}$$

We can therefore rewrite Equation (73):

$$AY = Y\Lambda \tag{75}$$

From $AY = Y\Lambda$ we can write:

$$\begin{aligned} P^{-1}AY &= P^{-1}Y\Lambda \\ \rightarrow P^{-1}APP^{-1}Y &= P^{-1}Y\Lambda \\ BP^{-1}Y &= P^{-1}Y\Lambda \\ BX &= X\Lambda \end{aligned} \tag{76}$$

We therefore know $X = P^{-1}Y$ stores the eigenvector of B.

From $(A - \Lambda)Y = 0$ we know it is satisfied that

$$\left(PBP^{-1} - \Lambda\right)Y = 0 \tag{77}$$

Therefore, B and A share the same eigenvalues stored in matrix Λ, with eigenvectors $P^{-1}Y$ for B. But note that this is only a partial proof, as (1) we shall not assume A is diagonalizable as it is not provided in the instructions, and (2) the diagonalizable A case may not be able to generalize to all cases.

One may also prove this without using the canonical decomposition (or a more general proof). From the definition, we may begin with

$$A\vec{v}_i = \lambda_i \vec{v}_i \tag{78}$$

One can further write:

$$P^{-1}A\vec{v}_i = P^{-1}\lambda_i \vec{v}_i \tag{79}$$

or can also be written in the form:

$$\left(P^{-1}A\right)\vec{v}_i = \lambda_i P^{-1}\vec{v}_i \tag{80}$$

Here, we may define that $P^{-1}\vec{v}_i = \vec{w}_i$ (from this we also know that $\vec{v}_i = P\vec{w}_i$). Equation (80) can be further rewritten as

$$P^{-1}AP\vec{w}_i = \lambda_i \vec{w}_i \tag{81}$$

We may interpret this equation from the geometric perspective, where the projection of matrix $P^{-1}AP$ on vector $\vec{w}_i$ is the same as the scalar multiplication by λ_i on vector $\vec{w}_i$. In other words, it writes:

$$\left(P^{-1}AP - \lambda_i\right)\vec{w}_i = 0 \tag{82}$$

where from this we know the vector $\vec{w}_i$ is in the nullspace of matrix $P^{-1}AP$. So $\vec{w}_i$ is an eigenvector of $P^{-1}AP$. Therefore, if we write $C = P^{-1}AP$, the equation

$$\left(C - \lambda_i\right)\vec{w}_i = 0 \tag{83}$$

says that λ_i is the eigenvalue of C. Hence, C and A have the same eigenvalues. We can then say $P^{-1}AP$ has the same eigenvalues as A. The statement is hence proved.

□

(d) *If D is a diagonal matrix, what are the eigenvalues of D?*

Solution. The eigenvalues would be the diagonal elements of D.

One can expand the characteristic equation to see this:

$$\begin{gathered}\det(D - \Lambda) = 0 \\ \rightarrow \begin{vmatrix} d_{11}-\lambda_1 & 0 & 0 & \dots & 0 \\ 0 & d_{22}-\lambda_2 & 0 & \dots & 0 \\ & & d_{33}-\lambda_3 & & \\ & & & \vdots & \\ & & & & d_{nn}-\lambda_n \end{vmatrix} = 0 \\ \prod_i^n (d_{ii} - \lambda_i) = 0 \end{gathered} \tag{84}$$

We therefore know that

$$\begin{cases} \lambda_1 = d_{11} \\ \lambda_2 = d_{22} \\ \lambda_3 = d_{33} \\ \vdots \\ \lambda_n = d_{nn} \end{cases} \tag{85}$$

So it is easy to see that the eigenvalues would be the diagonal elements, i.e., $\lambda_i = d_{ii}$. □

(e) *Consider the differential equation*

$$\frac{dx}{dt} = Ax.$$

Show that if $x(0)$ is an eigenvector of A with eigenvalue λ, then

$$x(t) = e^{\lambda t}x(0)$$

is a solution to the differential equation.

Solution. We may begin the proof by substituting $x(t) = e^{\lambda t}x(0)$ back to the ODE:

$$\begin{aligned} \frac{d\vec{x}}{dt} &= \frac{d}{dt}\left(e^{\lambda t}\vec{x}(0)\right) \\ \frac{d\vec{x}}{dt} &= \lambda e^{\lambda t}\vec{x}(0) + e^{\lambda t}\frac{d\vec{x}(0)}{dt} = A\vec{x} \end{aligned} \tag{86}$$

Since $x(0)$ is an eigenvector of A, we know

$$A\vec{x}(0) = \lambda\vec{x}(0) \tag{87}$$

Substitute this back to Equation (86) one has

$$\lambda e^{\lambda t}\vec{x}(0) + e^{\lambda t}\frac{d\vec{x}(0)}{dt} = \lambda e^{\lambda t}\vec{x}(0) \tag{88}$$

Since $x(0)$ is not a function of time, we know $\frac{d\vec{x}(0)}{dt} = 0$, therefore:

$$\lambda e^{\lambda t}\vec{x}(0) = \lambda e^{\lambda t}\vec{x}(0) \tag{89}$$

The relationship is hence established. Hence, one knows that $\vec{x}(t) = e^{\lambda t}\vec{x}(0)$ is a solution to the given ODE.

The statement is hence proved. □

ME300A HW#5

Hanfeng Zhai (hzhai@stanford.edu) December 9, 2023

Problem 1. *(Population Dynamics.) There are many different manners through which we can model population dynamics, but many of the models we use involve a system of ordinary differential equations. Let's start with a simple model.*

$$\frac{dP_1}{dt} = -0.8P_1 + 0.4P_2$$
$$\frac{dP_2}{dt} = -0.4P_1 + 0.2P_2$$

We start with a linear model for population dynamics, where P_1 represents the population of pandas (in thousands) and P_2 represents the population of bamboo caterpillars (in millions). The amount of bamboo eaten by pandas leads to them being heavy competitors within themselves as well as bamboo caterpillars for food. Caterpillars support their own population growth since they do not eat so much, but pandas will sometimes benefit from their population growth as an alternative food source.

1. *Write this linear system of differential equations as a matrix equation*

$$\frac{d\vec{P}}{dt} = A\vec{P},$$

where $\vec{P} = [P_1 \quad P_2]^T$. Identify the set of values for which the populations will be unchanging (i.e., fixed points, where $\frac{d\vec{P}}{dt} = 0$). What is the relationship between these values and the matrix A?

Solution. One can rewrite this linear system as

$$\begin{bmatrix} -0.8 & 0.4 \\ -0.4 & 0.2 \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} = \begin{bmatrix} \frac{dP_1}{dt} \\ \frac{dP_2}{dt} \end{bmatrix} \tag{1}$$

To find the fixed point, one needs to solve:

$$\begin{bmatrix} -0.8 & 0.4 \\ -0.4 & 0.2 \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{2}$$

Solving this linear system we have

$$2P_1 = P_2 \tag{3}$$

This indicates the general solution for the fixed point can be represented as

$$\vec{P} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} t, \quad t = \text{const.} \tag{4}$$

One can then substitute this back to the original matrix-vector multiplication and obtain the solution. Hence, vector P is a basis of the nullspace for matrix A. □

2. *Decouple (or diagonalize) A to write a general solution for $\vec{P}(t)$ with initial condition $\vec{P}(0)$. Is there a stable coexistence of a particular proportion of pandas and bamboo caterpillars? In other words, what happens to $P_1(t)$ and $P_2(t)$ as $t \to \infty$?*

   Hint: Recall that diagonalization allows us to express e^{At} as $Xe^{\Lambda t}X^{-1}$.

   Solution. The general solution writes

$$\begin{aligned} \vec{P} &= e^{At}\vec{P}(0) \\ &= Xe^{\Lambda t}X^{-1}\vec{P}(0) \\ \rightarrow \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} &= \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix} e^{\Lambda t} \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix}^{-1} \begin{bmatrix} P_1(0) \\ P_2(0) \end{bmatrix} \end{aligned} \tag{5}$$

To obtain X and Λ, one can solve for the eigenvectors and eigenvalues of A. For $\lambda_1 = 0$, one get the eigenvector

$$\vec{v}_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \tag{6}$$

For $\lambda_1 = -\frac{3}{5}$, one get the eigenvector

$$\vec{v}_2 = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \tag{7}$$

One can then use the normalized eigenvectors as a vector set:

$$\mathbf{V} = \left\{ \frac{1}{\sqrt{5}} \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \frac{1}{\sqrt{5}} \begin{bmatrix} 2 \\ 1 \end{bmatrix} \right\} \tag{8}$$

One can also write the eigenvalue matrix Λ:

$$\Lambda = \begin{bmatrix} 0 & 0 \\ 0 & -\frac{3}{5} \end{bmatrix} \tag{9}$$

Based on Λ and X (from $\mathbf{V}$), $A^{(t)}$ can be represented as

$$A^{(t)} = \begin{bmatrix} \frac{4\,\mathrm{e}^{-\frac{3\,t}{5}}}{3} - \frac{1}{3} & \frac{2}{3} - \frac{2\,\mathrm{e}^{-\frac{3\,t}{5}}}{3} \\ \frac{2\,\mathrm{e}^{-\frac{3\,t}{5}}}{3} - \frac{2}{3} & \frac{4}{3} - \frac{\mathrm{e}^{-\frac{3\,t}{5}}}{3} \end{bmatrix} \tag{10}$$

When $t \to \infty$, $A^{(t)}$ writes:

$$\lim_{t\to\infty} A^{(t)} = \frac{1}{3} \begin{bmatrix} -1 & 2 \\ -2 & 4 \end{bmatrix} \tag{11}$$

It can be observed that $P_1(t)$ and $P_2(t)$ agree with the general solution for the linear system of $\frac{d\vec{P}}{dt} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$. Here, if one were to determine the stable coexistence, we can substitute the initial condition back to the equation:

$$\begin{aligned} \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} &= Xe^{\Lambda t}X^{-1} \begin{bmatrix} P_1(0) \\ P_2(0) \end{bmatrix} \\ &= \lim_{t\to\infty} A^{(t)} \begin{bmatrix} P_1(0) \\ P_2(0) \end{bmatrix} \end{aligned} \tag{12}$$

$$= \frac{1}{3}\begin{bmatrix} -1 & 2 \\ -2 & 4 \end{bmatrix}\begin{bmatrix} P_1(0) \\ P_2(0) \end{bmatrix}$$

Since under the stable coexistence, the population of pandas and bamboo caterpillars should all be positive.

Hence, we can proceed with the equation

$$\begin{aligned} -P_1(0) + 2P_2(0) &> 0 \\ \rightarrow 2P_2(0) &> P_1(0) \end{aligned} \tag{13}$$

Which is the condition for the stable coexistence to exist for the equation. To be more precious (to answer the "in other words" in the instruction), both $P_1(t)$ and $P_2(t)$ are nonzero when $t \to \infty$ with the given initial condition.

□

This linear model was helpful for the first approach to modeling competitive species. Still, it would be nice if we could also model the effects of the limiting factor, the available bamboo. We adapt our model to include a new variable, B, which represents the bamboo population (in millions), and formulate a ***nonlinear*** *system of equations. We generalize the previous equation to include nonlinearity with $\frac{d\vec{P}}{dt} = \vec{f}(\vec{P})$.* ***Note: we have normalized all quantities so that reasonable populations should be $O(1)$.***

$$\begin{aligned} \frac{dP_1}{dt} &= -0.8P_1 + 0.4P_2 + 0.1P_1B \\ \frac{dP_2}{dt} &= -0.4P_1 + 0.2P_2 + 0.01P_2B^3 \\ \frac{dB}{dt} &= 1 - 0.1P_1 - 0.3P_2 - 0.25B \end{aligned}$$

1. *Write your own Newton-Raphson method in MATLAB to identify a positive fixed point (with elements all $O(1)$) for this system of equations and submit your code. Recall that for a multi-dimensional system, Newton-Raphson will generalize from 1D to multiple dimensions as:*

   $$\vec{x}^{(n+1)} = \vec{x}^{(n)} - J(\vec{x}^{(n)})^{-1}\vec{f}(\vec{x}^{(n)})$$

   where $J(\vec{x}^{(n)})$ is the Jacobian evaluated at $\vec{x} = \vec{x}^{(n)}$. Note that $J(\vec{x}^{(n)})$ will vary for each iteration, but you can calculate a formula for the Jacobian. Rather than construct the inverse of $J(\vec{x^{(n)}})$, we can save time by solving the linear system at every iteration:

   $$J(\vec{x}^{(n)})(\vec{x}^{(n+1)} - \vec{x}^{(n)}) = -\vec{f}(\vec{x}^{(n)})$$

   Feel free to use MATLAB's backslash \ operator to solve this linear system.

   Solution. Based on the nonlinear system:

   $$\vec{f} = \frac{d\vec{P}}{dt} \rightarrow \begin{cases} f_1 = \frac{dP_1}{dt} \\ f_2 = \frac{dP_2}{dt} \\ f_3 = \frac{dP_3}{dt} \end{cases} \tag{14}$$

with a solution vector $\vec{x} = \begin{bmatrix} P_1 \\ P_2 \\ B \end{bmatrix}$ One can thence expand the terms for the Jacobian:

$$J = \begin{bmatrix} -0.8 + 0.1B & 0.4 & 0.1P_1 \\ -0.4 & 0.2 + 0.01B^3 & 0.03P_2B^2 \\ -0.1 & -0.3 & -0.25 \end{bmatrix} \tag{15}$$

One can further expand the provided iteration scheme:

$$J(\vec{x}^{(n)}) \underbrace{\left(\Delta \vec{x}^{(n)}\right)}_{\vec{x}^{(n+1)} - \vec{x}^{(n)}} = -\vec{f}(\vec{x}^{(n)}) \tag{16}$$

And the target solution can then be obtained via solving the linear system

$$\left(\Delta \vec{x}^{(n)}\right) = -J^{-1}\vec{f} \tag{17}$$

Based on this simple formulation, one writes the following code, with a random initial vector $\vec{x}_0$ as $\vec{x}_0 = \begin{bmatrix} 0.1 \\ 0.1 \\ 0.1 \end{bmatrix}$:

```
x0 = [.1; .1; .1];
tolerance = 1e-10;
max_iter = 100;
iteration = 0;
while iteration < max_iter
    f_x = system_equations(x0);

    if norm(f_x) < tolerance
        fixed_point = x0;
        disp('Converged␣to␣a␣fixed␣point:');
        disp(fixed_point);
        return;
    end
    J_x = jacobian_matrix(x0);
    delta_x = J_x \ (-f_x);
    x0 = x0 + delta_x;
    iteration = iteration + 1;
end
```

With the corresponding functions write

```
function f_x = system_equations(x)
    f_x = [
        -0.8*x(1) + 0.4*x(2) + 0.1*x(1)*x(3);
        -0.4*x(1) + 0.2*x(2) + 0.01*x(2)*x(3)^3;
        1 - 0.1*x(1) - 0.3*x(2) - 0.25*x(3)
    ];
end
```

and

```
function J_x = jacobian_matrix(x)
    J_x = [
        -0.8 + 0.1*x(3), 0.4, 0.1*x(1);
        -0.4, 0.2 + 0.01*x(3)^3, 0.01*x(2)*3*x(3)^2;
        -0.1, -0.3, -0.25
    ];
end
```

And we get the converged solution from Newton-Raphson:

```
Converged to a fixed point:
    1.6854
    3.9836
   -1.4544
```

However, one should notice that here there is a negative fixed-point scenario, which should not be expected, considering we should not have a negative value of bamboo population. Hence, we can change the initial point and re-converge the iteration scheme. If one were to pick the initial point of $\vec{x}_0 = \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix}$, we converge to the fixed point:

$$\vec{P}_{fp} = \begin{bmatrix} 0.9749 \\ 1.5122 \\ 1.7954 \end{bmatrix} \tag{18}$$

which is in some sense correct. Because the bamboo population is positive (nonzero and not negative), with coexisting panda and caterpillar populations positive. Note that by testing a few other initial points verified the converged fixed point, e.g., $\vec{v}_0 = \begin{bmatrix} 1 \\ 3 \\ 2 \end{bmatrix}$, $\vec{v}_0 = \begin{bmatrix} 5 \\ 1 \\ 2 \end{bmatrix}$, $\vec{v}_0 = \begin{bmatrix} 1.2 \\ 5 \\ 1 \end{bmatrix}$, ...

We can then verify the accuracy of the convergence. Taking the $\begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix}$ as the initial point, we have

```
>> verify_fp = system_equations(fixed_point)

verify_fp =

   1.0e-13 *

   -0.8576
    0.3132
         0
```

indicating that the iteration indeed converges within the error tolerance. □

2. *Near the fixed point, we can approximate the behavior of the nonlinear system as something that looks like:*

$$\frac{d\vec{P}}{dt} = J(\vec{P}_{fp})\vec{P}$$

where $J(\vec{P}_{fp})$ is the Jacobian evaluated at the fixed point $\vec{P}_{fp}$. $J(\vec{P}_{fp})$ is then a constant coefficient matrix, meaning we have a ***linear*** *system of differential equations. Our situation is the same as the one we had in part (a), so we can decouple our system near this fixed point.*

Using MATLAB, identify the eigenvalues for this system. What do the real parts of the eigenvalues imply about the stability of the fixed point for long times?

Solution. Using MATLAB, one can evaluate the Jacobian at the fixed point to get $J(\vec{P}_{fp})$:

```
>> J_fp = jacobian_matrix(fixed_point)

J_fp =

   -0.6205    0.4000    0.0975
   -0.4000    0.2579    0.1462
   -0.1000   -0.3000   -0.2500
```

One can then get the eigenvector and eigenvalues of this coefficient matrix:

```
>> [v,d] = eig(J_fp)

v =

  -0.5109 + 0.0000i   0.1677 + 0.2761i   0.1677 - 0.2761i
  -0.1305 + 0.0000i   0.2705 + 0.4199i   0.2705 - 0.4199i
  -0.8497 + 0.0000i  -0.8038 + 0.0000i  -0.8038 + 0.0000i


d =

  -0.3562 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
   0.0000 + 0.0000i  -0.1282 + 0.1911i   0.0000 + 0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i  -0.1282 - 0.1911i
```

One can then get the real parts of the eigenvalues:

$$\begin{aligned} \lambda_1 &= -0.3562 \\ \lambda_2 &= -0.1282 \\ \lambda_3 &= -0.1282 \end{aligned} \tag{19}$$

We observe that all the real parts of the eigenvalues are negative. Since $\lim_{t\to\infty} e^{at} = 0$, implies the eigenvalues goes to zero. Hence, we can say this iteration scheme is stable. □

Problem 2. *(PageRank for Wikipedia.) In this question, we'll have a closer look at the PageRank algorithm. This algorithm famously invented for the Google search engine, is based on the idea that the most important websites will have many important websites linking to them. Here we will try applying the same algorithm to a data set of Wikipedia articles and the links between them.*

The PageRank algorithm can be formulated as a linear system:

$$\vec{x} = \alpha P \vec{x} + (1 - \alpha)\vec{v}$$

where the vector $\vec{x}$ describes the relative importance of a page, the "PageRank." The PageRank matrix P describes the linking structure between pages; in particular, P_{ij} can be thought of as the probability that page j links to page i when an outgoing link of j is taken at random. In other words, each column of P represents a probability vector describing the probability of transitioning from one page to all others. The vector $\vec{v}$ ascribes a base level of importance to all pages, and α is a positive scalar parameter that determines the amount of importance that propagates through links in the page network.

To simplify our problem, we will set $\alpha = 1$, so we are left with an eigenvalue equation for P, i.e. $\vec{x} = P\vec{x}$. The data set for this problem is sampled from a snapshot of English-language Wikipedia articles in 2023. Altogether the smaller data set we will work with contains the linking relationships between 10^5 of the webpages of Wikipedia.

To start, we will use an example 6 node case, with graph as in Fig. 1 and corresponding pagerank matrix:

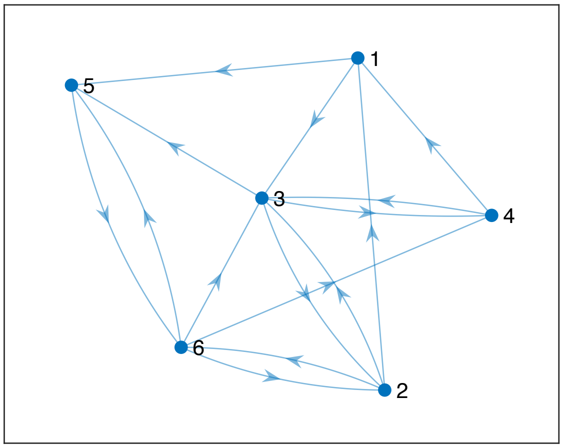


Figure 1: Directed graph for six webpages.

$$P = \begin{bmatrix} 0 & 0 & 0.25 & 0 & 0.333 & 0 \\ 0.5 & 0 & 0.25 & 0 & 0 & 0.5 \\ 0 & 0.5 & 0 & 0.5 & 0.333 & 0 \\ 0.5 & 0 & 0.25 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.5 \\ 0 & 0.5 & 0.25 & 0.5 & 0.333 & 0 \end{bmatrix}$$

1. *Write your own MATLAB function that implements the Power Method to determine the largest eigenvalue and eigenvector of any given PageRank matrix and submit your*

code. Using your favorite (nonzero) initial vector, apply it to the given PageRank matrix associated with the graph. What is the PageRank vector?

Solution. Based on the given iteration scheme, one can write the following MATLAB codes:

```
clc;clear
%%
P = [0 0 .25 0 .333 0;...
    .5 0 .25 0 0 .5;...
    0 .5 0 .5 .333 0;...
    .5 0 .25 0 0 0;...
    0 0 0 0 0 .5;...
    0 .5 .25 .5 .333 0];
%%
x_0 = [1 0 0 0 0]';
[D,k] = powermeth(P)
```

With the function writes:

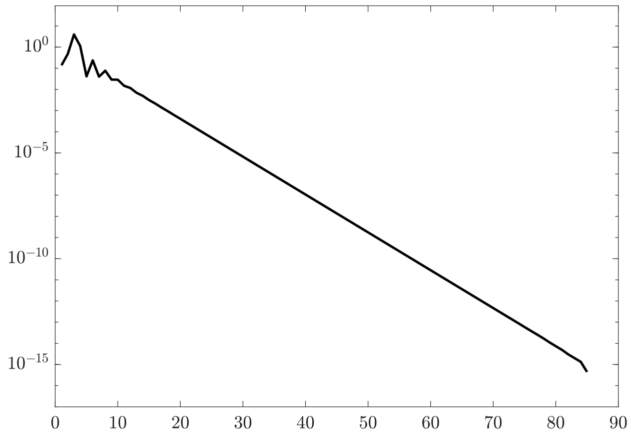
```
function [v,d,err] = powermeth(A)
         k = 1;  %initialize counter
         [n, n] = size(A);
         v = randn(n, 1);  % initialize with a random vector
         v = v / norm(v);
         d = v'*A*v;
         tol = 1e-15;
         max_iter = 10000;
         while k<max_iter

            v = A*v / norm(A*v);
            d_new = v'*A*v;
            err(k) = norm(d_new - d)/norm(d);
            if(norm(d_new - d)/norm(d) < tol)
                v = v / norm(v);
                d = d_new;
                break
            end
            d = d_new;
            k = k+1;

         end
end
```

In this implementation, my "favorite" initial vector is a randomized 1×6 vector: $\vec{x}_0 = \begin{bmatrix} 0.1001 \\ -0.5445 \\ 0.3035 \\ -0.6003 \\ 0.4900 \\ 0.7394 \end{bmatrix}$, and the iteration returned PageRank vector is $\vec{v} = \begin{bmatrix} 0.2134 \\ 0.5142 \\ 0.4656 \\ 0.2231 \\ 0.2911 \\ 0.5821 \end{bmatrix}$. Since the initial vectors are randomized each time, the algorithms converge to the same vector, verifying the correctness of the algorithm. □

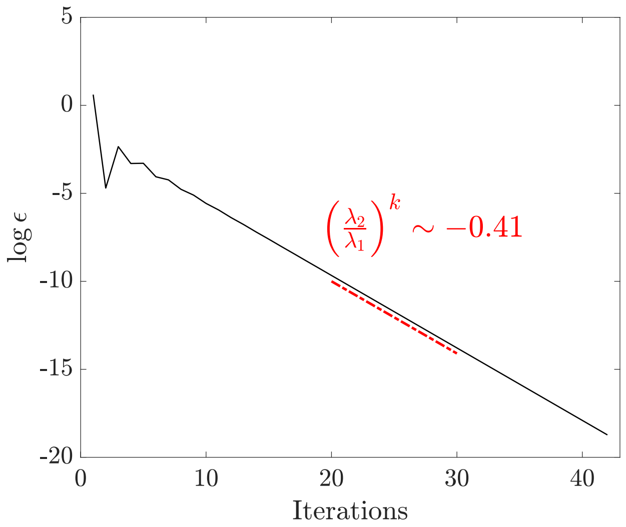
2. *For your Power Method function,* ***plot*** *the error norm against the iteration number on a* ***semilogy*** *plot.*
   Recall that the rate of convergence of the Power Method algorithm scales as $|\lambda_2/\lambda_1|^k$, where k is the iteration. Based on the slope of your error norm, what do you expect the magnitude of the next largest eigenvalue to be? Compare your prediction to the actual second largest eigenvalue in the magnitude of P using the ***eig*** *function.*

   Solution.

   By plotting using the “semilogy” we get the following figure:



The curve fitting procedure is shown as follows:



Based on the curve fit, one can solve this equation using a few lines of code:

```
syms lam2
eqn = abs(lam2/1)^2 == 0.41;
soln = solve(eqn, lam2); round(soln,3)
```

and obtain

```
ans =

0.64
```

Using the `eig` function, one obtains the magnitude of the second largest eigenvalues of P is 0.6624. It can then be deduced that our solution is 0.64 and the actual value is 0.6624, which is pretty close. The difference (~ 0.0224) is likely to be caused by the numerical precision of the computer. □

3. *We have provided two files, a sparse PageRank matrix for* 100, 000 *articles in* ***Pagerank_Transition.mat*** *and the names that correspond to each page in* ***Wikipedia_Article_Names.mat****. Use your algorithm to calculate the PageRank vector, and provide us with the top 10 Wikipedia articles and their corresponding PageRanks. Hint: Use both return values from the* ***sort*** *algorithm to retrieve both large values and corresponding indices.*

Solution. Using the provided data file, we use the power method and use the following codes:

```
clc;clear
load('Wikipedia_Article_Names.mat');
load('Pagerank_Transition.mat');
[v_trans,d_trans,err_trans] = powermeth(Transition_Probability_Matrix);
[sorted_ranks, indices] = sort(v_trans, 'descend');
top_10_indices = indices(1:10);
top_10_names = Article_Names(top_10_indices);
top_10_ranks = sorted_ranks(1:10);
```

The obtained top 10 articles are

```
>> top_10_names'

ans =

  10x1 cell array

    {'World␣War␣II'    }
    {'United␣States'   }
    {'Latin'           }
    {'Catholic␣Church'}
    {'United␣Kingdom'  }
    {'World␣War␣I'     }
    {'India'           }
    {'France'          }
    {'China'           }
    {'Soviet␣Union'    }
```

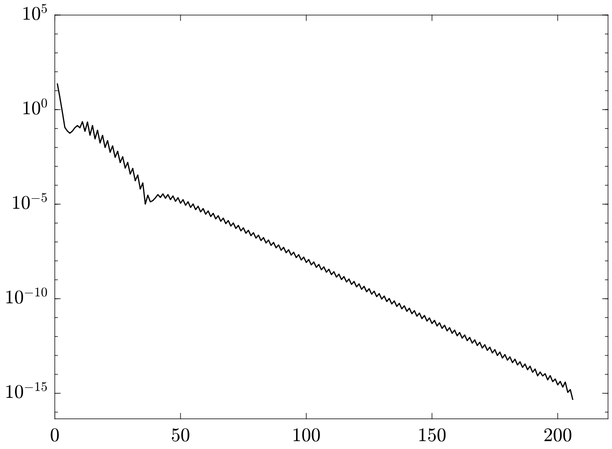
Their corresponding PageRanks are

```
>> top_10_ranks

top_10_ranks =

    0.1905
    0.1669
    0.1411
    0.1136
    0.1123
    0.1100
    0.0908
    0.0907
    0.0893
    0.0814
```

□

4. *Once again, plot the error norm against the iteration number to get a look at the convergence rate.*

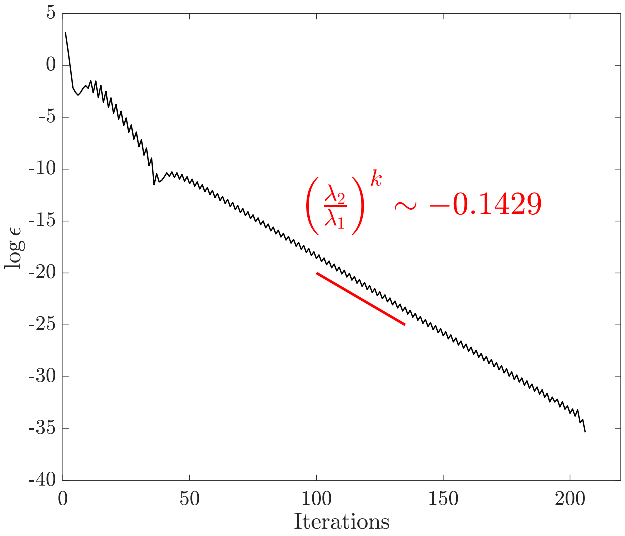
Solution. By plotting the convergence plot with **`semilogy`** method we generate the following figure:



Using a similar approach, one can also calculate the convergence rate by fitting the curve shown in the following figure. It can also be observed that in my implementation there are some "fluctuations" in the converging process. I attribute this "convergence fluctuation" to the numerical error caused by MATLAB.

Based on the set tolerance for this problem 10^{-15}, the power method converge to this tolerance after ~ 200 iterations.

□



5
0
-5
-10
-15
-20
-25
-30
-35
-40
$\log \epsilon$
0
50
100
150
200
Iterations
$\left(\frac{\lambda_2}{\lambda_1}\right)^k \sim -0.1429$