

# **ELEC60021 Mathematics For Signals and Systems**

Timothy Chung

Autumn 2023

# Contents

<b>1</b>	<b>Mathematical Background</b>	<b>3</b>
1.1	Systems of Linear equations . . . . .	3
1.2	Gaussian Elimination . . . . .	6
1.3	LU Decomposition . . . . .	8
1.4	Vector Spaces, Inner Product and Norm . . . . .	10
1.5	Cauchy-Schwarz Inequality . . . . .	12
1.5.1	Introduction . . . . .	12
1.5.2	Proof of the Cauchy-Schwarz Inequality . . . . .	12
1.6	Linear Mappings, Basis and Dimension . . . . .	13
1.6.1	Span . . . . .	13
1.6.2	Basis and Dimension . . . . .	13
1.6.3	Orthogonal (Unitary) Matrices . . . . .	13
1.7	Applications . . . . .	14
1.7.1	Linear Mappings and Matrices . . . . .	14
1.7.2	Understanding Matrix Operations through Linear Mappings . . . . .	14
1.7.3	Signal Processing . . . . .	14
1.7.4	Circulant Matrices . . . . .	16
1.7.5	Discrete-Time Convolution and FIR Filters . . . . .	17
1.7.6	Linear and Circular Convolution . . . . .	17
1.7.7	Discrete-time Fourier Transform . . . . .	19
1.8	Range and Null Space . . . . .	19
1.8.1	Introduction . . . . .	19
1.8.2	Computing the Nullspace . . . . .	21
<b>2</b>	<b>Matrix Theory Fundamentals</b>	<b>23</b>
2.1	Determinant . . . . .	23
2.2	Rank and Trace . . . . .	24
2.3	Eigenvectors and Eigenvalues . . . . .	24
2.3.1	Eigenvectors and Eigenvalues of a permutation matrix . . . . .	25
2.3.2	Solving for Eigenvalues and Eigenvectors . . . . .	25
2.3.3	Special Case: Eigenvalues with Identity Matrix Addition . . . . .	26
2.4	Diagonalisation of Matrices . . . . .	26
2.4.1	Matrix Diagonalisation and Commutativity . . . . .	27
2.5	Difference Equations . . . . .	28
2.6	Markov Processes . . . . .	29
2.7	Diagonalisation of Circulant Matrices . . . . .	30
2.8	Complex, Symmetric, Hermitian Matrices . . . . .	31
2.8.1	Spectral Theorem . . . . .	32
2.8.2	Positive Definite Hermitian Matrices . . . . .	33
2.9	Graph Adjacency Matrix . . . . .	33
2.10	Pagerank Algorithm . . . . .	34
<b>3</b>	<b>Solving <math>Ax = b</math></b>	<b>37</b>
3.1	Interpretations . . . . .	37
3.2	Existence and Uniqueness . . . . .	38
3.3	Four Fundamental Subspaces . . . . .	39
3.4	Inverses . . . . .	41
3.5	Projections and Approximations . . . . .	41

3.6	Gram-Schmidt Orthogonalisation Process . . . . .	42
3.7	Solutions, either exact or approximate . . . . .	43
3.7.1	Example: Least-Square Deconvolution . . . . .	44
3.7.2	Example: High-Order Regression . . . . .	45
3.7.3	Solution (Full Row-Rank Case) . . . . .	45
3.8	Computing Left and Right Inverses . . . . .	46
3.9	Least-Square Problems with Linear Constraints . . . . .	47
<b>4</b>	<b>Pseudoinverses and the SVD</b>	<b>48</b>
4.1	Singular Value Decomposition . . . . .	48
4.1.1	Diagonalization of a Matrix . . . . .	48
4.1.2	The Singular Value Decomposition Theorem . . . . .	48
4.1.3	Properties of SVD . . . . .	48
4.1.4	More properties of the SVD . . . . .	49
4.2	Relating the SVD to the pseudo-inverse . . . . .	50
4.3	Applications of the SVD . . . . .	50
4.3.1	Low Rank Approximation . . . . .	50
4.3.2	Principal Component Analysis . . . . .	51
4.3.3	Total Least Squares . . . . .	52
4.3.4	Kaczmarz Algorithm . . . . .	54
4.4	Summary . . . . .	55

“Linear Algebra = Money” – Prof Pier-Luigi Dragotti

# Chapter 1

## Mathematical Background

### 1.1 Systems of Linear equations

Consider a system of two linear equations with two unknowns:

$$2x - y = 0 \quad (1.1)$$

$$-x + 2y = 3 \quad (1.2)$$

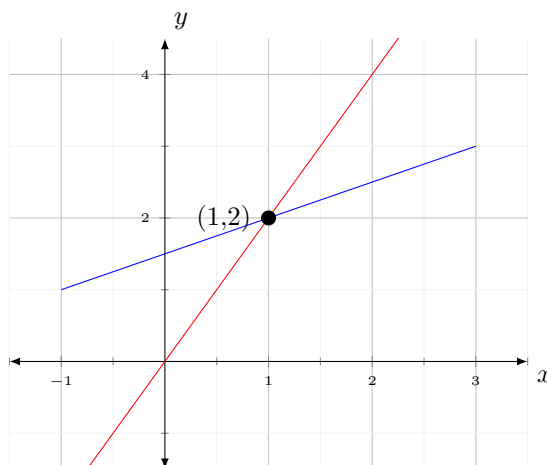
This system can be represented in matrix form as:

$$\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \end{bmatrix}$$

where the matrix on the left is the coefficient matrix, and the vector on the right is the vector of constants.

#### Graphical Interpretation

Each equation in the system corresponds to a straight line in the 2D plane, and the solution of the system is the intersection point of these lines.



This graphical representation is known as the **Row Formulation** of the system.

#### Column Formulation

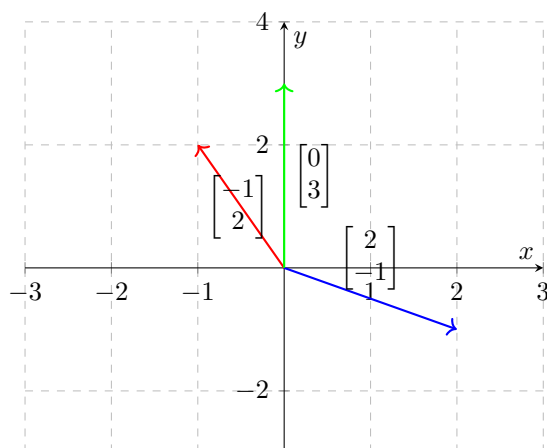
Alternatively, we can express the system in terms of a linear combination of column vectors, known as the **Column Formulation**.

$$x \begin{bmatrix} 2 \\ -1 \end{bmatrix} + y \begin{bmatrix} -1 \\ 2 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \end{bmatrix}$$

The weights  $x$  and  $y$  are the scalars that, when used to multiply the respective column vectors and then added together, yield the resultant vector on the right-hand side.

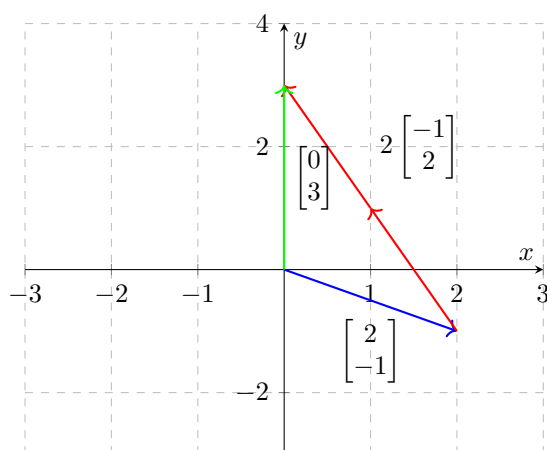
## Geometrical Representation

The solution to the system of equations corresponds to the specific linear combination of the column vectors that results in the right-hand side vector.



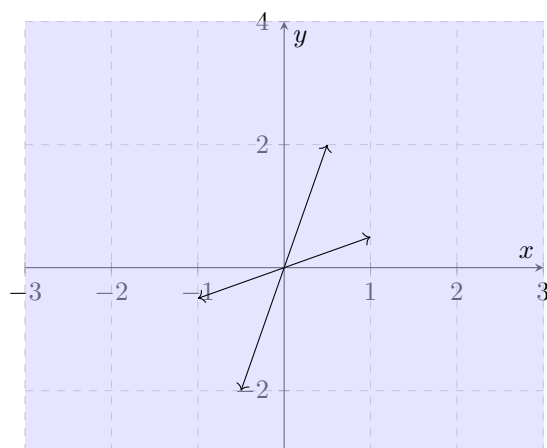
The combination of  $x = 1$  and  $y = 2$  yields the solution to the system:

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



## Span of Vectors

The span of a set of vectors is the collection of all possible linear combinations of those vectors. In the case of two non-parallel vectors in 2D space, their span is the entire plane.



This illustrates that all possible linear combinations of the columns  $\begin{bmatrix} 2 \\ -1 \end{bmatrix}$  and  $\begin{bmatrix} -1 \\ 2 \end{bmatrix}$  cover the entire 2D plane.

## Matrix Formulation

In matrix formulation we abandon scalar variables and numbers. Every entity is part of either a matrix or a vector as follows:

$$\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \end{bmatrix}$$

We assume we have  $m$  equations and  $n$  unknowns and can depict the matrix formulation as:

$$\mathbf{Ax} = \mathbf{b}$$

where  $\mathbf{A}$  is a matrix of size  $m \times n$  and  $\mathbf{x}$  is a column vector of size  $m \times 1$ .

## 3D Matrix Example

Consider the following system of three equations with three unknowns:

$$2x - y = 0, \tag{1.3}$$

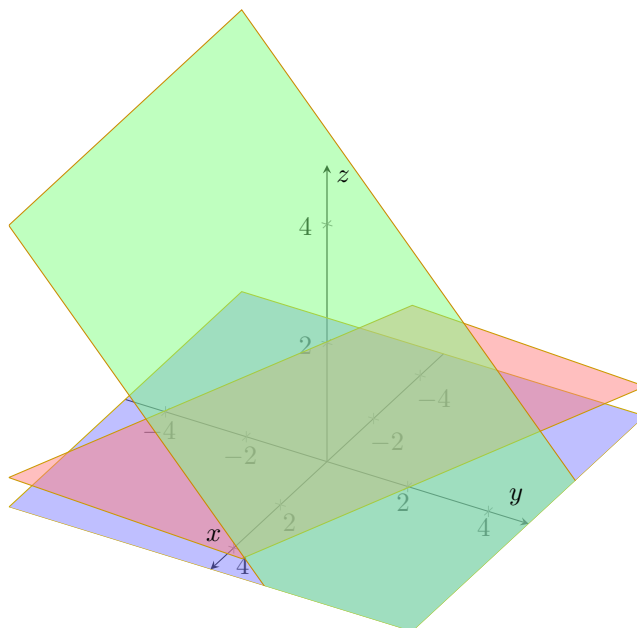
$$-x + 2y - z = -1, \tag{1.4}$$

$$-3y + 4z = 4. \tag{1.5}$$

This system can be written in matrix form  $\mathbf{Ax} = \mathbf{b}$  as:

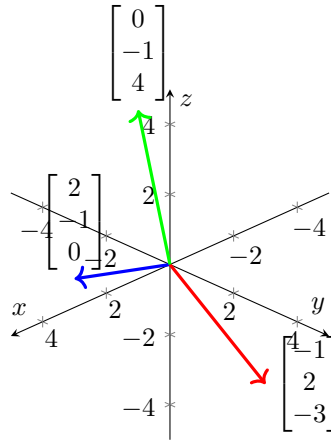
$$\mathbf{A} = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -3 & 4 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 0 \\ -1 \\ 4 \end{bmatrix}.$$

In row formulation, each row represents a plane in the 3D space. The solution of the system is where these three planes intersect. Visualising the solution in higher dimensions can be challenging.



In column formulation, the system is viewed as a combination of column vectors scaled by the unknowns:

$$x \begin{bmatrix} 2 \\ -1 \\ 0 \end{bmatrix} + y \begin{bmatrix} -1 \\ 2 \\ -3 \end{bmatrix} + z \begin{bmatrix} 0 \\ -1 \\ 4 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \\ 4 \end{bmatrix}.$$



The solution is the particular combination of column vectors that yield the right-hand side vector. For the given system, the solution is  $\mathbf{x} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$ .

- The solutions of a system of three equations with three unknowns lie within the 3D space.
- A matrix  $\mathbf{A}$  is invertible if its column vectors are linearly independent, meaning they do not lie in the same plane.
- If at least one column is a linear combination of the others, the matrix is not invertible, and the system may not have a unique solution for every  $\mathbf{b}$  in  $\mathbb{R}^3$ .

The concept of row and column formulations leads to the broader idea of vector spaces. Before delving deeper into vector spaces, it is essential to understand practical algorithms to solve systems of linear equations, such as Gaussian elimination.

## 1.2 Gaussian Elimination

Gaussian Elimination, also known as Row Reduction, is a systematic method for solving systems of linear equations. It involves performing operations on the rows of the augmented matrix of the system to obtain a row-echelon form, from which the solutions can easily be determined through back-substitution.

### Example System

Consider the system of linear equations:

$$x + 2y + z = 2, \quad (1.6)$$

$$3x + 8y + z = 12, \quad (1.7)$$

$$4y + z = 2. \quad (1.8)$$

### Row Reduction Process

**Step 1:** Eliminate  $x$  from the second and third equations.

We multiply the first equation by 3 and subtract it from the second to eliminate  $x$ :

$$\begin{aligned} [2] - 3[1] : \quad & (3x + 8y + z) - 3(x + 2y + z) = 12 - 6, \\ & 2y - 2z = 6. \end{aligned}$$

**Step 2:** Multiply the first equation by 2 and subtract it from the third to eliminate  $x$ :

$$\begin{aligned} [3] - 2[1] : \quad & (4y + z) - 2(x + 2y + z) = 2 - 2(2), \\ & 5z = -10. \end{aligned}$$

**Step 3:** Now, we have a new system of equations:

$$x + 2y + z = 2, \quad (1.9)$$

$$2y - 2z = 6, \quad (1.10)$$

$$5z = -10. \quad (1.11)$$

**Step 4:** We have achieved an upper triangular matrix.



## Back-Substitution

Once the matrix is in upper triangular form, we can find the solutions by back-substitution. The final form of the matrix and the solutions are:

$$x + 2y + z = 2, \quad (1.12)$$

$$2y - 2z = 6, \quad (1.13)$$

$$5z = -10, \quad (1.14)$$

Since the system of equations is now in upper triangular form, we can solve the equations in reverse order. This method is straightforward because each equation can be solved for one variable at a time, starting from the last equation.

We start by solving the last equation for  $z$ :

$$5z = -10$$

$$z = \frac{-10}{5}$$

$$z = -2.$$

Substituting  $z$  into the second equation, we solve for  $y$ :

$$2y - 2(-2) = 6$$

$$2y + 4 = 6$$

$$2y = 6 - 4$$

$$2y = 2$$

$$y = \frac{2}{2}$$

$$y = 1.$$

Finally, we substitute  $y$  and  $z$  back into the first equation to solve for  $x$ :

$$x + 2(1) - 2 = 2$$

$$x + 2 - 2 = 2$$

$$x = 2 - 2 + 2$$

$$x = 2.$$

The solution to the system is:

$$x = 2,$$

$$y = 1,$$

$$z = -2.$$

We solve the equations in reverse order because the system after elimination is triangular, allowing us to express each variable in terms of the subsequent ones.

## Matrix Transformation

The original matrix  $\mathbf{A}$  is:

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 1 \\ 3 & 8 & 1 \\ 0 & 4 & 1 \end{bmatrix}.$$

After applying the row operations  $[2] - 3[1]$  and  $[3] - 2[2]$ , the matrix is transformed into:

$$\mathbf{U} = \begin{bmatrix} \boxed{1} & 2 & 1 \\ 0 & \boxed{2} & -2 \\ 0 & 0 & \boxed{5} \end{bmatrix},$$

where the elements in the diagonal of  $\mathbf{U}$  are called *pivots*.

## Column Transformation

Similarly, the column vector  $\mathbf{b}$  is transformed alongside matrix  $\mathbf{A}$  as follows:

$$\mathbf{b} = \begin{bmatrix} 2 \\ 12 \\ 2 \end{bmatrix} \rightarrow \begin{bmatrix} 2 \\ 6 \\ -10 \end{bmatrix}.$$

## Augmented Matrix

It is often convenient to combine the matrix  $\mathbf{A}$  and the vector  $\mathbf{b}$  into an augmented matrix  $[\mathbf{A}|\mathbf{b}]$  and perform row operations on this combined matrix:

$$[\mathbf{A}|\mathbf{b}] = \left[ \begin{array}{ccc|c} 1 & 2 & 1 & 2 \\ 3 & 8 & 1 & 12 \\ 0 & 4 & 1 & 2 \end{array} \right] \rightarrow \left[ \begin{array}{ccc|c} 1 & 2 & 1 & 2 \\ 0 & 2 & -2 & 6 \\ 0 & 0 & 5 & -10 \end{array} \right].$$

This augmented matrix represents the system after the row operations have been applied, and it is now ready for back-substitution to solve for the unknowns.

## 1.3 LU Decomposition

The process of Gaussian Elimination can be represented algebraically using matrices. Specifically, each elementary row operation can be performed by multiplying the matrix of coefficients by an *elimination matrix*.

### Transformation of Matrix $\mathbf{A}$

Consider the system of linear equations represented in matrix form by  $\mathbf{Ax} = \mathbf{b}$ . The coefficient matrix  $\mathbf{A}$  is transformed through the elimination process as follows:

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 1 \\ 3 & 8 & 1 \\ 0 & 4 & 1 \end{bmatrix} \xrightarrow{[2] - 3[1]} \begin{bmatrix} 1 & 2 & 1 \\ 0 & 2 & -2 \\ 0 & 4 & 1 \end{bmatrix} \xrightarrow{[3] - 2[2]} \begin{bmatrix} 1 & 2 & 1 \\ 0 & 2 & -2 \\ 0 & 0 & 5 \end{bmatrix} = \mathbf{U}$$

### Elimination Matrices

Each step where a substitution of the form  $[i] + c \times [j]$  is performed corresponds to multiplying the current matrix by an identity matrix with its  $i, j$  element replaced by  $c$ . This special matrix is denoted  $E_{ij}$ .

For instance, the first step  $[2] - 3[1]$  can be represented by an elimination matrix  $E_{21}$ :

$$E_{21} = \begin{bmatrix} 1 & 0 & 0 \\ -3 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

### Matrix Multiplication Representation

The second step of elimination,  $[3] - 2[2]$ , can be represented as:

$$E_{32} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{bmatrix}$$

Applying these elimination matrices to  $\mathbf{A}$  sequentially gives us  $\mathbf{U}$ :

$$E_{32}(E_{21}\mathbf{A}) = \mathbf{U}$$

The brackets can be dropped due to the associative property of matrix multiplication, thus:

$$E_{32}E_{21}\mathbf{A} = \mathbf{U}$$

## Inverses of Elimination Matrices

The inverse of an elimination matrix  $E_{ij}$  is conveniently obtained by changing the sign of its non-zero off-diagonal element. For example:

$$E_{21}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad E_{32}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 2 & 1 \end{bmatrix}$$

The inverses of these matrices reverse the elimination steps, providing a method to return to the original matrix  $\mathbf{A}$ . LU decomposition is a method where a matrix  $\mathbf{A}$  is factorized into two matrices,  $\mathbf{L}$  and  $\mathbf{U}$ , where  $\mathbf{L}$  is a lower triangular matrix and  $\mathbf{U}$  is an upper triangular matrix. This factorization is particularly useful for solving systems of equations, inverting matrices, and computing determinants.

## Matrix Factorization

The elimination process can be expressed as a sequence of matrix multiplications, transforming the original matrix  $\mathbf{A}$  into an upper triangular matrix  $\mathbf{U}$ :

$$E_{32}E_{21}\mathbf{A} = \mathbf{U}$$

## Inversion of Elimination Matrices

By sequentially multiplying both sides from the left with the inverses of the elimination matrices in reverse order, we can express  $\mathbf{A}$  as:

$$\mathbf{A} = E_{21}^{-1}E_{32}^{-1}\mathbf{U}$$

## Constructing Matrix $\mathbf{L}$

The product of the inverses of the elimination matrices gives us matrix  $\mathbf{L}$ :

$$\mathbf{L} = E_{21}^{-1}E_{32}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ 0 & 2 & 1 \end{bmatrix}$$

Matrix  $\mathbf{L}$  has the nice property that its elements below the diagonal are the multipliers used in the elimination process.

## LU Decomposition

Thus, matrix  $\mathbf{A}$  can be decomposed as:

$$\mathbf{A} = \mathbf{L}\mathbf{U}$$

This is the LU decomposition of  $\mathbf{A}$ .

## Permutation Matrix

In the general case where row exchanges are required during the elimination process, we introduce a permutation matrix  $\mathbf{P}$ . The  $\mathbf{PA} = \mathbf{LU}$  factorization incorporates these permutations:

$$\mathbf{PA} = \mathbf{LU}$$

A permutation matrix  $\mathbf{P}$  arises from the identity matrix if we reorder the rows to ensure that the pivot elements are non-zero.

## Example of a Permutation Matrix

For instance, if we need to swap rows 1 and 2 in matrix  $\mathbf{A}$  to get a non-zero pivot in the first position, the permutation matrix  $\mathbf{P}_{12}$  and the operation would look like this:

$$\mathbf{P}_{12} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{P}_{12}\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 \\ 3 & 8 & 1 \\ 0 & 4 & 1 \end{bmatrix} = \begin{bmatrix} 3 & 8 & 1 \\ 1 & 2 & 1 \\ 0 & 4 & 1 \end{bmatrix}$$

This reordering is essential to avoid dividing by zero when attempting to use that pivot to eliminate the variables in the column below.

For any invertible matrix  $\mathbf{A}$ , the LU decomposition with partial pivoting (where row exchanges are accounted for) can be expressed as:

$$\mathbf{PA} = \mathbf{LU}$$

Here,  $\mathbf{P}$  is the permutation matrix that records the row exchanges,  $\mathbf{L}$  is the product of the inverses of the elimination matrices (lower triangular), and  $\mathbf{U}$  is the resulting upper triangular matrix after applying Gaussian elimination.

The complete LU decomposition with a permutation matrix is thus represented as:

$$\mathbf{A} = \mathbf{P}^{-1}\mathbf{LU}$$

It is important to note that  $\mathbf{P}^{-1} = \mathbf{P}^T$  since permutation matrices are orthogonal.

## 1.4 Vector Spaces, Inner Product and Norm

We need to define a common framework to handle vectors in dimensions  $\leq 3$  (geometry is useful) and in dimensions  $> 3$ .

A vector space is a collection of vectors that form a nonempty set  $E$  with two operations: addition and scalar multiplication that satisfy the following properties:

- **Commutativity:** For all  $x, y \in E$ , we have  $x + y = y + x$ .
- **Associativity:** For all  $x, y, z \in E$ , we have  $(x + y) + z = x + (y + z)$ .
- **Distributivity:** For all  $x \in E$  and scalars  $\alpha, \beta$ , we have  $(\alpha + \beta)x = \alpha x + \beta x$  and  $\alpha(x + y) = \alpha x + \alpha y$ .
- **Existence of Inverses:** For every  $x, y \in E$  there exists  $z \in E$  such that  $x + z = y$ .
- **Compatibility of Scalars and Multiplication:** For scalars  $\alpha, \beta$  and all  $x \in E$ , we have  $\alpha(\beta x) = (\alpha\beta)x$ .
- **Identity Element:** There exists an identity element  $1$  such that for every  $x \in E$ , we have  $1x = x$ .

Let  $E$  be a complex vector space. A mapping  $\langle \cdot, \cdot \rangle$  from  $E \times E \rightarrow \mathbb{C}$  is called an inner product in  $E$  if for any  $x, y, z \in E$  and  $\alpha, \beta \in \mathbb{C}$  the following conditions are satisfied:

- **Conjugate Symmetry:**  $\langle x, y \rangle = \overline{\langle y, x \rangle}$ .
- **Linearity in the first argument:**  $\langle \alpha x + \beta y, z \rangle = \alpha \langle x, z \rangle + \beta \langle y, z \rangle$ .
- **Positive-definiteness:**  $\langle x, x \rangle \geq 0$  and  $\langle x, x \rangle = 0$  implies  $x = 0$ .

In  $\mathbb{C}$ , the standard inner product between components of vectors  $a, b \in \mathbb{C}^n$  can be written as

$$\sum_{i=1}^n a_i \bar{b}_i \quad (1.15)$$

Where  $\bar{b}_i$  is the complex conjugate of  $b_i$ .

It is always useful to be able to determine the size of an object, in the case of vector spaces this is achieved using the notion of ‘norm’.

A norm on a vector space  $E$  over  $\mathbb{C}$  (or  $\mathbb{R}$ ) is a real-valued function with the following properties for any  $x, y \in E$  and  $\alpha \in \mathbb{C}$ :

- **Positive definiteness:**  $\|x\| \geq 0$ , and  $\|x\| = 0$  if and only if  $x = 0$ .
- **Homogeneity Property:**  $\|\alpha x\| = |\alpha| \|x\|$ .

- **Triangle Inequality:**  $\|x + y\| \leq \|x\| + \|y\|$ .

Given the definition of inner product, we define the induced norm of a vector  $x$  as:

$$\|x\|_2 = \sqrt{\langle x, x \rangle} = \sqrt{\sum_{i=1}^n |x_i|^2}$$

The more general norm is defined as:

$$\|x\|_p = \sqrt[p]{\langle x, x \rangle} = \left( \sum_{i=1}^n |x_i|^p \right)^{\frac{1}{p}}$$

## Examples of Vector Spaces

### n-dimensional Vectors in $\mathbb{R}^n$ or $\mathbb{C}^n$

A vector is a set of scalars placed jointly in a vertical fashion (column vector).

- A column vector of size  $n$  is denoted as  $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$ .
- The Hermitian transpose of  $\mathbf{x}$  is a row vector of size  $n$  denoted as  $\mathbf{x}^H = (x_1^* \ x_2^* \ \cdots \ x_n^*)$ .
- Addition and product with a scalar are easily defined, while the inner product is:  $\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{y}^H \mathbf{x} = \sum_{i=1}^n x_i y_i^*$ .
- Consequently, the induced norm is:  $\|\mathbf{x}\| = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle} = \left( \sum_{i=1}^n |x_i|^2 \right)^{1/2}$ .

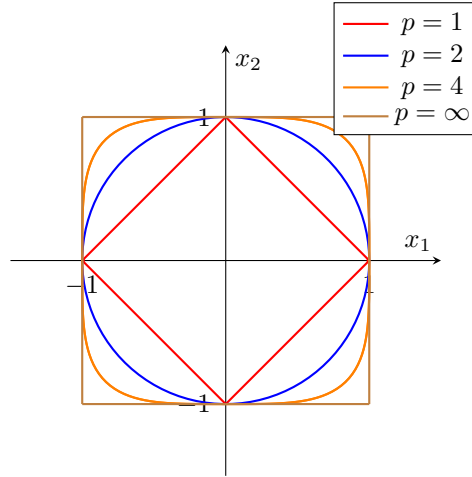
### Matrices of size $m \times n$

The space of matrices of size  $m \times n$ , that is, the space with elements  $A \in \mathbb{C}^{m \times n}$  includes:

- Matrices are typically used to model linear equations but are more complex objects.
- The inner product of two matrices  $A, B$  is the element-by-element product:  $\langle A, B \rangle = \sum_{i,j} a_{i,j} b_{i,j}^*$ .
- This leads to the induced norm (Frobenius norm):  $\|A\|_F = \sqrt{\sum_{i,j} |a_{i,j}|^2}$ .
- More general norms:  $\|A\|_p = \sup_{\|x\|_p \neq 0} \frac{\|Ax\|_p}{\|x\|_p}$ .

# On the Notion of the Norm

## Unit-norm Vectors for Different p-norms



Note: In this plot, the  $p$ -norm is defined for a unit vector  $x$  in  $\mathbb{R}^2$  as  $\|x\|_p = (|x_1|^p + |x_2|^p)^{1/p}$ , where  $p$  takes values of 1, 2, 4,  $\infty$ , for the plotted shapes. Note that  $p$  cannot take on non-integer values.

## 1.5 Cauchy-Schwarz Inequality

The Cauchy-Schwarz Inequality is a fundamental result in vector analysis which provides a bound for the absolute value of the inner product of two vectors.

### 1.5.1 Introduction

For all vectors  $\mathbf{x}, \mathbf{y}$  in an inner-product space, the following inequality holds:

$$|\langle \mathbf{x}, \mathbf{y} \rangle| \leq \|\mathbf{x}\| \|\mathbf{y}\|$$

This inequality allows us to estimate the similarity of two vectors and to introduce the idea of direction. It is particularly useful in defining the angle between two vectors:

$$\cos \theta = \frac{\langle \mathbf{x}, \mathbf{y} \rangle}{\|\mathbf{x}\| \|\mathbf{y}\|}$$

Two vectors are considered 'maximally' similar when one is a rescaled version of the other, and they are 'maximally' dissimilar, or orthogonal, when  $\theta = \frac{\pi}{2}$ , that is, when  $\langle \mathbf{x}, \mathbf{y} \rangle = 0$ . If the vectors are orthogonal and their magnitudes are 1, they are called orthonormal.

### 1.5.2 Proof of the Cauchy-Schwarz Inequality

*Sketch of the proof:* Starting from the non-negativity of the norm, we have:

$$0 \leq \|\mathbf{x} - \alpha \mathbf{y}\|^2 = \langle \mathbf{x} - \alpha \mathbf{y}, \mathbf{x} - \alpha \mathbf{y} \rangle$$

Expanding the inner product, we get:

$$= \|\mathbf{x}\|^2 - \langle \mathbf{x}, \alpha \mathbf{y} \rangle - \langle \alpha \mathbf{y}, \mathbf{x} \rangle + |\alpha|^2 \|\mathbf{y}\|^2$$

Choosing  $\alpha = \frac{\langle \mathbf{x}, \mathbf{y} \rangle}{\|\mathbf{y}\|^2}$  and using the property that if  $c = \langle \mathbf{x}, \mathbf{y} \rangle$  then  $\langle \mathbf{y}, \mathbf{x} \rangle = c^*$ , we obtain:

$$0 \leq \|\mathbf{x}\|^2 - \frac{|\langle \mathbf{x}, \mathbf{y} \rangle|^2}{\|\mathbf{y}\|^2}$$

Which simplifies to the Cauchy-Schwarz inequality:

$$|\langle \mathbf{x}, \mathbf{y} \rangle|^2 \leq \|\mathbf{x}\|^2 \|\mathbf{y}\|^2$$

And taking the square root of both sides gives us the final form of the inequality.

## 1.6 Linear Mappings, Basis and Dimension

Let  $v_1, v_2, \dots, v_n$  be vectors in a vector space of dimension  $n$ . For scalar coefficients  $c_i \in \mathbb{R}$ , the linear combination  $x$  can be expressed as:

$$x = c_1 v_1 + c_2 v_2 + \dots + c_n v_n$$

In matrix/vector form, this is represented as:

$$x = \begin{bmatrix} v_{1,1} & v_{1,2} & \cdots & v_{1,n} \\ v_{2,1} & v_{2,2} & \cdots & v_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ v_{n,1} & v_{n,2} & \cdots & v_{n,n} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}$$

- The vectors  $v_1, v_2, \dots, v_n$  are **independent** if no linear combination of them gives the zero vector (except the trivial combination where all  $c_i = 0$ ).
- Two non-zero, non-parallel vectors in two-dimensional space are independent because no scalar multiples of one can be added to obtain the other.
- Considering three vectors in the two-dimensional space, they are **dependent** if there exists non-trivial scalars  $x_i$  such that:

$$x_1 v_1 + x_2 v_2 + x_3 v_3 = 0$$

This means at least one vector is a linear combination of the others.

### 1.6.1 Span

Span	Definition 1.6.1
<p>Let <math>T</math> be a set of vectors in a vector space <math>E</math>. The <b>span</b> of <math>T</math>, denoted <math>\text{span}(T)</math>, is the set of all vectors that can be expressed as linear combinations of the vectors in <math>T</math>. Formally, for <math>V = \text{span}(T)</math>, any vector <math>x \in V</math> can be written as:</p> $x = \sum_i c_i v_i$ <p>where <math>c_i \in \mathbb{R}</math> and <math>v_i \in T</math>. By construction, <math>V</math> is itself a vector space, and since <math>V \subseteq E</math>, it is a subspace of <math>E</math>.</p>	

**Example:** The span of two linearly independent vectors in  $\mathbb{R}^3$  forms a plane, which is a subspace of  $\mathbb{R}^3$ .

*Question:* Is the following plane a subspace as well?

$$V = \{x \in \mathbb{R}^3 : x_1 + x_2 + x_3 = 1\}$$

To be a subspace,  $V$  must include the zero vector and be closed under addition and scalar multiplication. Since the sum of the components must equal 1,  $V$  does not include the zero vector and thus is not a subspace.

### 1.6.2 Basis and Dimension

Vector Space Basis	Definition 1.6.2
<p>A <i>basis</i> for a vector space <math>E</math> is a set of vectors <math>T</math> that are linearly independent and span <math>E</math>. The number of vectors in the basis is called the <i>dimension</i> of <math>E</math>.</p>	

**Example:** The standard basis for  $\mathbb{R}^n$  or  $\mathbb{C}^n$  consists of vectors  $e_i$ , where each  $e_i$  has a 1 in the  $i$ -th position and 0s elsewhere. This basis is known as the *canonical basis*.

When a basis also satisfies  $\langle v_i, v_j \rangle = 0$  for  $i \neq j$ , it is called an *orthogonal basis*. If additionally  $\langle v_i, v_i \rangle = 1$ , it is an *orthonormal basis*.

**Exercise:** Consider  $S = \{(1, 2, \gamma)^T : \gamma \in \mathbb{R}\} \subset \mathbb{R}^3$ . The dimension of  $\text{span}(S)$  is 2, as  $S$  can be expressed as the span of  $\{(1, 2, 0)^T, (0, 0, 1)^T\}$ , which are linearly independent.

### 1.6.3 Orthogonal (Unitary) Matrices

An *orthonormal basis* of  $n$ -dimensional space can be formed into a matrix  $A$  that is orthogonal (unitary in the complex case). Such a matrix preserves the inner product and, consequently, the induced norm.

### Properties of Orthogonal Matrices:

- For matrix  $A$  with orthonormal vectors as columns,  $A^H A = A A^H = I$ , where  $I$  is the identity matrix, indicating  $A^H = A^{-1}$ .
- Orthogonal matrices preserve the induced norm:  $\|Ax\|_2 = \|x\|_2$ .
- The process to orthogonalize a set of vectors is known as the *Gram-Schmidt process*.

### Examples of Orthogonal Matrices

**Rotation Matrices:** A rotation matrix in 2D is defined by  $Q = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$ , which satisfies  $QQ^T = Q^T Q = I$ .

**Permutation Matrices:** Permutation matrices, such as  $P = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ , reorder the rows of the identity matrix. They satisfy  $PQ = QP = I$ .

### Orthogonal Subspaces

Extending the concept of orthogonality to subspaces requires every vector in one subspace to be orthogonal to every vector in the other subspace.

Orthogonality	Definition 1.6.3
Two subspaces $V$ and $W$ of $E$ are orthogonal if every vector $v \in V$ is orthogonal to every vector $w \in W$ : $\langle v, w \rangle = 0$ .	
Subspaces	Definition 1.6.4
The <i>orthogonal complement</i> to a subspace $V \subset E$ , denoted $V^\perp$ , is the set of all vectors in $E$ that are orthogonal to every vector in $V$ .	

## 1.7 Applications

### 1.7.1 Linear Mappings and Matrices

- We initially thought of matrices and vectors as a means to express linear equations in a compact form. Now, with a deeper understanding of vector spaces, we aim to interpret matrices more ‘geometrically’.
- Matrices facilitate the description of linear transformations between vector spaces. These transformations map vectors from one space, known as the domain, to another, known as the range.

### 1.7.2 Understanding Matrix Operations through Linear Mappings

Assume vector spaces  $X, Y \in \mathbb{R}^2$ , and consider a linear mapping represented by matrix  $A$  such that  $y = Ax$  with:

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

This matrix  $A$  corresponds to a rotation by an angle  $\theta$  in the 2-dimensional Euclidean space.

- To describe a rigid rotation in 3-D, we use rotation matrices that depend on the axis of rotation. For example, rotation about the  $z$ -axis by an angle  $\theta$  is represented by a similar trigonometric matrix with the addition of a third dimension.
- A 2-D rotation by  $2\theta$  can be achieved by multiplying two matrices of rotation by  $\theta$ , exploiting the property that matrix multiplication is associative.
- To invert a 2-D rotation by  $\theta$ , we can either transpose the rotation matrix (since rotation matrices are orthogonal) or construct a new rotation matrix with angle  $-\theta$ , effectively rotating in the opposite direction.

### 1.7.3 Signal Processing

Many important operations in signal processing can be describe using matrices.



### Time-Reversal of Signals

Given a discrete-time signal  $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$ , its time-reversed version  $\mathbf{y} = (x_n, \dots, x_2, x_1)^T$  can be described by:

$$\mathbf{y} = R\mathbf{x}$$

where  $R$  is the time-reversal matrix, a square matrix with 1's on its anti-diagonal and 0's elsewhere:

$$R = \begin{bmatrix} 0 & \cdots & 0 & 1 \\ 0 & \cdots & 1 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 1 & \cdots & 0 & 0 \end{bmatrix} \quad R\mathbf{x} = \begin{bmatrix} 0 & \cdots & 0 & 1 \\ 0 & \cdots & 1 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 1 & \cdots & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} x_n \\ x_{n-1} \\ \vdots \\ x_1 \end{bmatrix}$$

### Truncation of Signals

Given a discrete-time signal  $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$ , to truncate the last  $m - n$  terms of a signal with  $m < n$ , use a truncation matrix  $T$  of size  $m \times n$  with 1's on the diagonal (the left part is a  $m \times m$  identity matrix) and 0's elsewhere:

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

Then  $\mathbf{y} = T\mathbf{x}$ .

$$\mathbf{y} = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \\ x_{m+1} \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}$$

### Zero-Padding Operation

Zero-padding a signal  $\mathbf{x}$  to length  $m > n$  can be done using an extension matrix  $E$  of size  $m \times n$ . The zero-padded signal is  $\mathbf{y} = E\mathbf{x}$ .

$$E = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} \quad E\mathbf{x} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \\ 0_{n+1} \\ \vdots \\ 0_m \end{bmatrix}$$

### Permutation Operation

A permutation matrix  $P$  rearranges the elements of a vector  $\mathbf{x}$ . For example, to swap the first and second elements:

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

Applying  $P$  to  $\mathbf{x}$  yields the permuted vector. The permutation matrix has a single '1' in every row and column.

## Differentiation of Polynomials

We can represent a general polynomial of degree  $n$  with a  $n+1$ -dimensional vector. For example,  $p(x) = c_0 + c_1x + c_2x^2$  can be represented as  $\mathbf{p} = \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix}$ . The derivative of  $p(x)$  is  $p'(x) = c_1 + 2c_2x$  represented as  $\mathbf{p}' = \begin{bmatrix} c_1 \\ 2c_2 \\ 0 \end{bmatrix}$ . We can

express the differentiation using matrix  $D_2 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 2 \\ 0 & 0 & 0 \end{bmatrix}$  that satisfies  $D_2\mathbf{p} = \mathbf{p}'$

For differentiating a polynomial of degree  $d$ , represented by a vector of its coefficients  $\mathbf{a}$ , the differentiation matrix  $D$  is:

$$D = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & d \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}$$

The derivative  $\mathbf{a}'$  is obtained by  $\mathbf{a}' = D\mathbf{a}$ .

### 1.7.4 Circulant Matrices

Circulant matrices are a special class of matrices that play an important role in various applications, including signal processing and time series analysis. They are defined by the property that each row is a cyclic shift of the one above it.

#### Shifting a Vector

Assume we have a vector  $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$ . To shift this vector by one unit to the right (or when looking vertically, one unit to the bottom), we can use a permutation matrix  $P$ . For  $n = 4$ , the matrix and the shifted vector are given by:

$$P = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad P\mathbf{x} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} x_4 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Multiplying  $\mathbf{x}$  by  $P^2$  and  $P^3$  will cyclically shift the vector by 2 and 3 units, respectively.

#### Constructing a Circulant Matrix

A circulant matrix  $C$  is constructed as a linear combination of identity matrix  $I$  and powers of the permutation matrix  $P$ . For  $n = 4$ ,  $C$  can be expressed as:

$$C = c_0I + c_1P + c_2P^2 + c_3P^3 = \begin{bmatrix} c_0 & c_3 & c_2 & c_1 \\ c_1 & c_0 & c_3 & c_2 \\ c_2 & c_1 & c_0 & c_3 \\ c_3 & c_2 & c_1 & c_0 \end{bmatrix}$$

Importantly, this structure enables  $C$  to perform a filtering operation, where the  $c_i$ 's represent the filter coefficients.

$$\mathbf{y} = C\mathbf{x}$$

The action of  $C$  can be thought of as applying a filter to the signal  $\mathbf{x}$ , where the filter characteristics are determined by the coefficients  $c_i$ . More generally, it can be defined as:

$$C = \begin{bmatrix} c_0 & c_{n-1} & \cdots & c_2 & c_1 \\ c_1 & c_0 & c_{n-1} & \cdots & c_2 \\ \vdots & c_1 & c_0 & \ddots & \vdots \\ c_{n-2} & \cdots & \ddots & \ddots & c_{n-1} \\ c_{n-1} & c_{n-2} & \cdots & c_1 & c_0 \end{bmatrix} \quad (1.16)$$

Convolution is a process of combining two sequences to produce a third sequence, which is the modification or filtering of an original signal.

## Convolution through Circulant Matrices

Multiplying a circulant matrix  $C$  by a vector  $x$  is equivalent to convolving the first row of  $C$  with  $x$ . For example:

$$C = \begin{bmatrix} c_0 & c_1 & c_2 \\ c_2 & c_0 & c_1 \\ c_1 & c_2 & c_0 \end{bmatrix}$$

$$\mathbf{x} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix}$$

$$\mathbf{y} = C\mathbf{x} = \begin{bmatrix} c_0x_0 + c_1x_1 + c_2x_2 \\ c_2x_0 + c_0x_1 + c_1x_2 \\ c_1x_0 + c_2x_1 + c_0x_2 \end{bmatrix}$$

### 1.7.5 Discrete-Time Convolution and FIR Filters

Discrete-time convolution is a fundamental operation in signal processing where a filter with an impulse response  $h_k$  is applied to an input signal  $x_n$  to produce an output signal  $y_n$ . This process is mathematically expressed as:

$$y_n = \sum_k h_k x_{n-k}$$

The operation involves 'sliding' the filter  $h_k$  over the input sequence  $x_n$  and summing the products of  $h_k$  and  $x_{n-k}$  for each  $n$ . It is assumed that the filter has a finite impulse response (FIR), meaning that  $h_k \neq 0$  only for a finite number of samples  $k$ , and that the input sequence also has finite duration where  $n > K$ .

Under such conditions, convolution is a linear mapping from  $\mathbb{R}^n$  to  $\mathbb{R}^m$ :

$$\mathbf{y} = A\mathbf{x}$$

Convolution can be represented as a linear operation using matrices. For FIR filters, this operation is typically represented by a Toeplitz matrix  $A$ , which has constant values along its diagonals. The matrix multiplication  $y = Ax$  can then be used to perform the convolution, where  $A$  is structured such that each row is a shifted version of the impulse response  $h_k$ .

$A \in \mathbb{R}^{m \times n}$  is generally represented as:

$$y = h * x = A\mathbf{x} = \begin{bmatrix} h_0 & 0 & \cdots & 0 & 0 \\ h_1 & h_0 & & \vdots & \vdots \\ h_2 & h_1 & \cdots & 0 & 0 \\ \vdots & h_2 & \cdots & h_0 & 0 \\ h_{K-2} & \vdots & \ddots & h_1 & h_0 \\ h_{K-1} & h_{K-2} & & \vdots & h_1 \\ 0 & h_{K-1} & \ddots & h_{K-3} & \vdots \\ 0 & 0 & \cdots & h_{K-2} & h_{K-3} \\ \vdots & \vdots & & h_{K-1} & h_{K-2} \\ 0 & 0 & 0 & \cdots & h_{K-1} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix}$$

### 1.7.6 Linear and Circular Convolution

The difference between linear and circular convolution is distinguished by how the input signal is treated. Linear convolution, which uses a Toeplitz matrix, assumes the input  $x$  is non-periodic and has a length  $m = n + K - 1$  to accommodate the length of the convolved signal. In contrast, circular convolution assumes the input signal is periodic and wraps around the end of the signal back to the start. This is modeled using a circulant matrix  $A$ , which is a special case of the Toeplitz matrix where the end of the impulse response wraps around to the beginning of the next row.

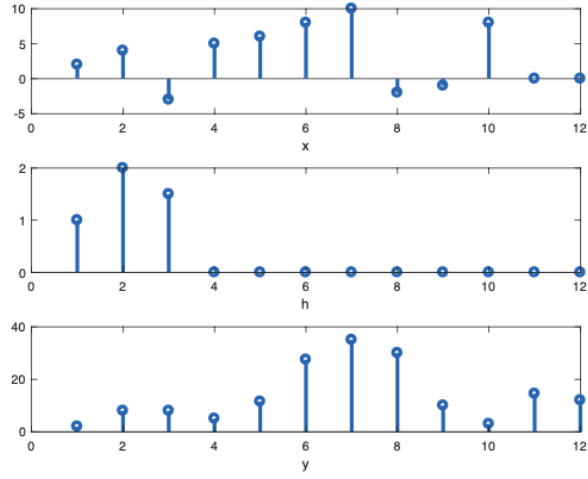


Figure 1.1:  $A \in \mathbb{R}^{m \times n}$  with  $m = n + K - 1$  models the linear convolution.

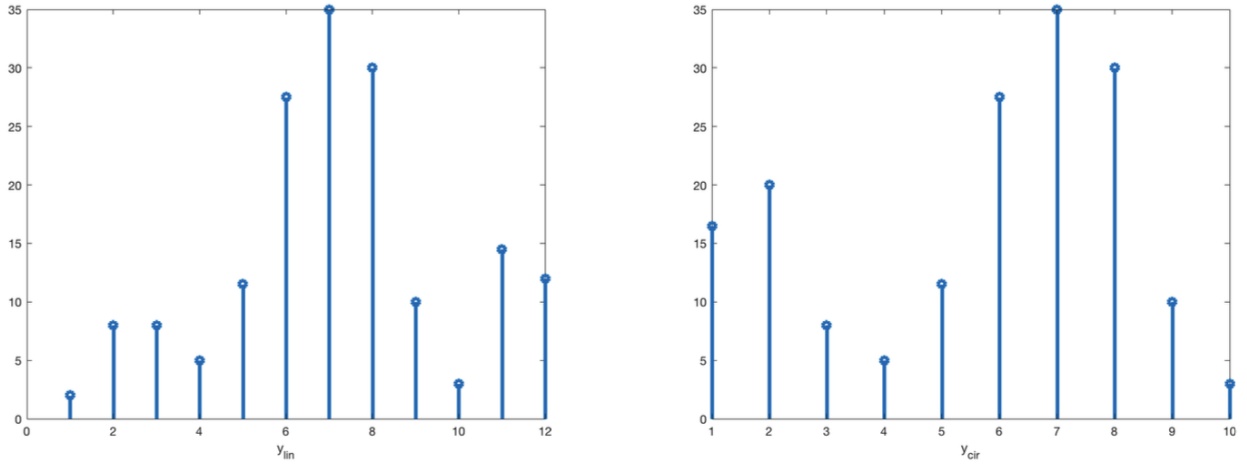


Figure 1.2: If we want  $A$  to be square, we may assume that  $x \in \mathbb{R}^n$  is periodic outside the interval of observation. In this case  $A$  is a ‘circulant’ matrix and models the circulant convolution.

### Example of Linear and Circulant Convolution

Assume you want to implement a discrete-time version of the derivative  $y(t) = \frac{dx(t)}{dt}$ . The natural way to do it is by taking finite differences of the incoming sequence:  $y_n = x_n - x_{n-1}$ . This is achieved by filtering with  $h_k = [1, -1]$ :

$$y_n = \sum_k h_k x_{n-k}$$

In the case of linear convolution, the linear mapping is described by the  $(n + 1) \times n$  matrix:

$$\mathbf{A} = \begin{bmatrix} h_0 & 0 & \cdots & 0 & 0 \\ h_1 & h_0 & & \vdots & \vdots \\ h_2 & h_1 & \cdots & 0 & 0 \\ \vdots & h_2 & \cdots & h_0 & 0 \\ h_{K-2} & \vdots & \ddots & h_1 & h_0 \\ h_{K-1} & h_{K-2} & & \vdots & h_1 \\ 0 & h_{K-1} & \ddots & h_{K-3} & \vdots \\ 0 & 0 & \cdots & h_{K-2} & h_{K-3} \\ \vdots & \vdots & & h_{K-1} & h_{K-2} \\ 0 & 0 & 0 & \cdots & h_{K-1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ -1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & -1 & 1 & 0 \\ 0 & 0 & \cdots & 0 & -1 & 1 \\ 0 & 0 & \cdots & 0 & 0 & -1 \end{bmatrix}$$

In the case of circulant convolution, the linear mapping is described by the  $n \times n$  matrix:

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ -1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & -1 & 1 & 0 \\ 0 & 0 & \cdots & 0 & -1 & 1 \end{bmatrix}$$

### 1.7.7 Discrete-time Fourier Transform

The discrete Fourier Transform (DFT) is a linear mapping and can be described by a matrix. Recall that given the finite-length sequence:  $x_0, x_1, \dots, x_{N-1}$  the DF is given by:

$$X_r = \sum_{n=0}^{N-1} x_n e^{-j2\pi \frac{rn}{N}} \quad r = 0, 1, \dots, N-1$$

$$\begin{bmatrix} X_0 \\ X_1 \\ \vdots \\ X_{N-1} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & e^{-j\frac{2\pi}{N}} & e^{-j\frac{4\pi}{N}} & \vdots & e^{-j2\pi \frac{N-1}{N}} \\ 1 & e^{-j\frac{4\pi}{N}} & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & e^{-j2\pi \frac{(N-2)(N-1)}{N}} \\ 1 & e^{-j2\pi \frac{N-1}{N}} & \cdots & e^{-j2\pi \frac{(N-1)(N-2)}{N}} & e^{-j2\pi \frac{(N-1)(N-1)}{N}} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ \vdots \\ x_{N-1} \end{bmatrix}$$

## 1.8 Range and Null Space

### 1.8.1 Introduction

Consider the linear transformation  $\mathbf{X} \rightarrow \mathbf{Y}$  represented by the matrix

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

The matrix  $A$  maps vectors from  $\mathbb{R}^3$  to  $\mathbb{R}^3$ . However, the first two rows of  $A$  are identical, indicating that all vectors  $\mathbf{X}$  are mapped to a plane in  $\mathbb{R}^3$ . This redundancy suggests that the mapping is not one-to-one since different vectors in  $\mathbb{R}^3$  could result in the same vector in  $\mathbb{R}^3$  after transformation. Consequently, we infer that the transformation is not invertible.

In an intuitive sense, the elements have been collapse into a 2D plane despite the mapping being from  $\mathbb{R}^3$  to  $\mathbb{R}^3$ . Since information 'is lost' after transformation, we feel that this mapping is not invertible.

The concepts of range space (or column space) and null space (or kernel) of  $A$  help to formalise this intuition. The range space is the set of all possible outputs  $\mathbf{Y}$ , which, in this case, is a plane in  $\mathbb{R}^3$ . The null space is the set of all

vectors  $\mathbf{X}$  that map to the zero vector in  $\mathbb{R}^3$ , indicating the loss of information and confirming the non-invertibility of the transformation.

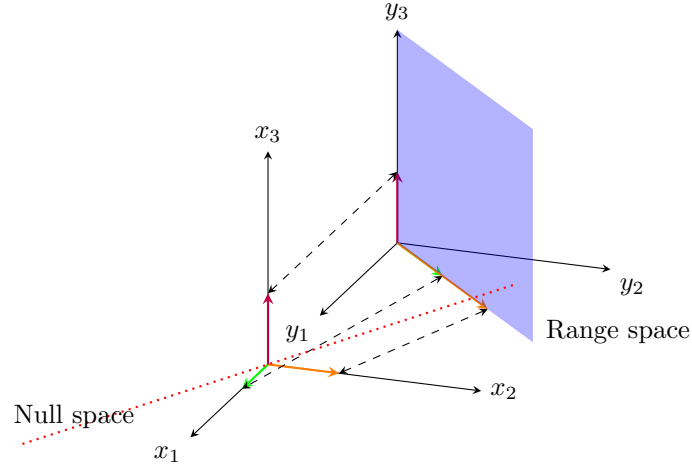


Figure 1.3: Visualised mapping

We observe:

- The columns of  $\mathbf{A}$  are linearly dependent.
- The first and third column are only independent columns.
- The plane where all the vectors  $\mathbf{x}$  were collapsing to is the span $\left\{\begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}\right\}$ , the range of the matrix
- We note that if  $\mathbf{x} = \begin{pmatrix} 2 \\ -1 \\ 0 \end{pmatrix}$  then  $\mathbf{Ax} = 0$ , so  $\mathbf{x} = \begin{pmatrix} 2 \\ -1 \\ 0 \end{pmatrix}$  is in the null space of  $\mathbf{A}$  and is orthogonal to the range space.

#### Range and Null Space

#### Definition 1.8.1

Let  $A : X \rightarrow Y$  be a linear transformation. The **range space**  $\mathcal{R}(A)$  is the set of values in  $Y$  that are reached from  $X$  by applying  $A$ :

$$\mathcal{R}(A) = \{y = Ax : x \in X\}$$

The **null space**  $\mathcal{N}(A)$  is the set of values in  $X$  that are transformed to 0 by  $A$ :

$$\mathcal{N}(A) = \{x \in X : Ax = 0\}$$

Let  $A$  be an  $m \times n$  matrix which we regard as a linear operator and which we write as follows:

$$A = [v_1 \quad v_2 \quad \dots \quad v_n]$$

then a vector  $x \in \mathbb{R}^n$  is transformed as:

$$Ax = x_1v_1 + x_2v_2 + \dots + x_nv_n$$

which is a linear combination of the columns of  $A$ .

This means that the range can be expressed as

$$\mathcal{R}(A) = \text{span}\{v_1, v_2, \dots, v_n\}$$

and corresponds to the column space of  $A$ . Moreover, the number of linear independent columns of  $A$ , which by definition is also the rank of  $A$ , corresponds to the dimension of  $\mathcal{R}(A)$ .

The null space of  $A$  is also a subspace and its dimension is  $n - \text{rank}(A)$ .

When the dimension of  $\mathcal{N}(A) > 0$  then the mapping is not invertible. Assume that  $An_1 = 0$  and that  $y_1 = Ax_1$ , then, given  $x_2 = x_1 + n_1$ , we have that  $Ax_2 = Ax_1 + An_1 = Ax_1 = y_1$ . Therefore given  $y_1$  it is not possible to know whether it is due to  $x_1$  or  $x_2$ .

#### Sketch of the Proof:

- Remember that when we compute  $Ax$  we are effectively taking a linear combination of the columns of  $A$
- Assume for the sake of argument that  $\text{rank}(A) = n - 2$ , this means that given  $n - 2$  columns of  $A$ , the other two columns can be obtained with proper linear combinations of the linearly independent  $n - 2$  columns.
- Therefore there are two linearly independent vectors  $x_1$  and  $x_2$  such that  $Ax_1 = Ax_2 = 0$
- Moreover any linear combination of  $x_1$  and  $x_2$  is also in the null space of  $A$ , therefore that space has dimension 2.

Assume that  $A$  models a linear mapping from  $\mathbb{C}^n$  to  $\mathbb{C}^n$  and that its null space is non-trivial (i.e., it has dimension larger than 0 or  $\text{rank}(A) < n$ ) then the mapping is not invertible or in other words the matrix  $A$  does not have an inverse.

In this case we call  $A$  a **singular matrix**.

### 1.8.2 Computing the Nullspace

- Get row-echelon form (staircase)  $u$
- Identify pivot columns and free columns
- To find nullspace solve  $ux = 0$
- Find the solution expressed as a function of free variables that correspond to the pivot columns position

Let's consider matrix  $A$  and calculate its null space.

Given the matrix:

$$A = \begin{bmatrix} 1 & 2 & 2 & 2 \\ 2 & 4 & 6 & 8 \\ 3 & 6 & 8 & 10 \end{bmatrix}$$

By applying elimination, we obtained the matrix  $u$ :

$$u = \begin{bmatrix} 1 & 2 & 2 & 2 \\ 0 & 0 & 2 & 4 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

The matrix  $u$  contains two **pivot columns** shown in red and two **free columns** shown in blue.

To calculate the null space  $\mathcal{N}(A)$ , we need to solve the system  $Ax = 0$ . The system  $Ax = 0$  is equivalent to  $ux = 0$  which can be written as:

$$ux = \begin{bmatrix} 1 & 2 & 2 & 2 \\ 0 & 0 & 2 & 4 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = 0$$

Using row formulation, we obtain:

$$\begin{aligned} x_1 + 2x_2 + 2x_3 + 2x_4 &= 0 \\ 2x_3 + 4x_4 &= 0 \end{aligned}$$

The solution of the above system is of the form:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} -2x_2 + 2x_4 \\ x_2 \\ -2x_4 \\ x_4 \end{bmatrix} = x_2 \begin{bmatrix} -2 \\ 1 \\ 0 \\ 0 \end{bmatrix} + x_4 \begin{bmatrix} 2 \\ 0 \\ -2 \\ 1 \end{bmatrix}$$

Variables  $x_2$  and  $x_4$  can take any values (free variables).

By assigning the value of 1 to a particular free variable and the value of 0 to the rest of the free variables, we obtain a so-called **special solution**.

The first special solution is obtained for  $x_2 = 1, x_4 = 0$  and is:

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

The second special solution is obtained for  $x_2 = 0, x_4 = 1$  and is:

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

The null space is the linear combination of the special solutions:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = c \begin{bmatrix} -2 \\ 1 \\ 0 \\ 0 \end{bmatrix} + d \begin{bmatrix} 2 \\ 0 \\ -2 \\ 1 \end{bmatrix}$$

## Range and Null Space for Linear Filtering

- Convolution and linear filtering are fundamental operations in signal processing. They can be algebraically represented using matrices.
- For linear systems, the convolution operation can be described by the Toeplitz matrix  $A$  through the equation  $y = Ax$ , where  $y$  is the output signal and  $x$  is the input signal.
- The process of deconvolution aims to reverse this operation, attempting to retrieve the input signal  $x$  from the output signal  $y$ .
- Understanding the range (or column space) and the null space of the matrix  $A$  is crucial because it informs us about the invertibility of the convolution operation. Specifically, the range tells us what outputs are possible, and the null space indicates if there are any inputs that could yield a zero output.
- **Claim** (without proof): If matrix  $A$  represents a linear convolution, then the null space is typically trivial, meaning there are no non-zero inputs that give a zero output. This is in contrast to circulant convolution, where the null space is generally non-trivial, allowing for the existence of non-zero inputs that do produce a zero output.
- **Example:** In the context of finite difference operations, consider a vector  $x$  consisting entirely of ones. For a circulant convolution matrix  $A$ , this particular  $x$  would lie in the null space of  $A$ , meaning that applying  $A$  to  $x$  would yield a zero vector, which signifies that no change would be detected in the signal.



## Chapter 2

# Matrix Theory Fundamentals

### 2.1 Determinant

The Determinant is a crucial number associated with square matrices, denoted either by  $\det(A)$  or  $|A|$ . A matrix  $A$  is invertible iff (if and only if)  $\det(A) \neq 0$ .

Invertible matrix is called **non-singular**. For a 2x2 matrix  $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$  the determinant is defined as  $\begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc$ .

Laplace's Formula for the Determinant	Definition 2.1.1
<p>The determinant is defined as:  <math>\det(\mathbf{A}) = \sum_{j=1}^n (-1)^{i+j} a_{ij} M_{ij}</math> for fixed <math>i</math> or  <math>\det(\mathbf{A}) = \sum_{i=1}^n (-1)^{i+j} a_{ij} M_{ij}</math> for fixed <math>j</math>            This summation within can just be shortened to the product of the cofactor and minor matrices within <math>\mathbf{A}</math>, where:</p> <ul style="list-style-type: none"> <li>The <i>minor</i> <math>M_{i,j}</math> is defined to be the determinant of the <math>(n-1) \times (n-1)</math>-matrix that results from <math>A</math> by removing the <math>i</math>-th row and the <math>j</math>-th column.</li> <li>The <i>cofactor</i> <math>C_{i,j}</math> is obtained by multiplying the minor by <math>(-1)^{i+j}</math>.</li> </ul>	

#### Properties of Determinants

1.  $\det(I) = 1$
2. Exchanging two rows of a matrix reverses the sign of the determinant. So an even number of exchanges keeps the determinant unchanged and an odd number of exchanges inverts the sign.
3. Multiplying any row with a scalar multiplies the determinant by that scalar.
4. Any row of zeroes leads to  $\det = 0$ .

$$\begin{vmatrix} 0 & 0 \\ c & d \end{vmatrix} = \begin{vmatrix} 0 \cdot a & 0 \cdot b \\ c & d \end{vmatrix} = 0 \cdot \begin{vmatrix} a & b \\ c & d \end{vmatrix} = 0$$

5. Note that the determinant of the sum of two matrices is not necessarily the sum of their determinants:

$$\begin{vmatrix} a+a' & b+b' \\ c & d \end{vmatrix} = \left( \begin{vmatrix} a & b' \\ c & d \end{vmatrix} + \begin{vmatrix} a' & b \\ c & d \end{vmatrix} \right) \neq \begin{vmatrix} a & b \\ c & d \end{vmatrix} + \begin{vmatrix} a' & b' \\ c & d \end{vmatrix}$$

In other words,  $\det(A+B) \neq \det(A) + \det(B)$ . We observe 'linearity' only for a single row

6. Two equal rows leads to  $\det = 0$ . This is inferred from the case of exchanging rows that changes the sign of the determinant – if we exchange two same rows, the final matrix is the same, so it should have the same sign – the only possible case of this is when  $\det = 0$ .
7. Determinant after row reduction remains the same.

$$\begin{vmatrix} a & b \\ c-la & d-lb \end{vmatrix} = \begin{vmatrix} a & b \\ c & d \end{vmatrix} + \begin{vmatrix} a & b \\ -la & -lb \end{vmatrix} = \begin{vmatrix} a & b \\ c & d \end{vmatrix} - l \begin{vmatrix} a & b \\ a & b \end{vmatrix} = \begin{vmatrix} a & b \\ c & d \end{vmatrix}$$

8. Any upper or lower triangular matrix has the determinant that is the product of its diagonals.
9. Any singular matrix has  $\det = 0$  because eliminating the matrix gives it a row of zero. So this matrix is not full rank and is invertible. Any invertible matrix will have a non-zero determinant.
10. For square matrices  $\mathbf{A}$  and  $\mathbf{G}$  we have  $\det(\mathbf{AB}) = \det(\mathbf{A})\det(\mathbf{B})$
11. Moreover:
  - The determinant of the inverse of  $A$  is the reciprocal of the determinant of  $A$ :

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

- The determinant of  $A$  squared is the square of the determinant of  $A$ :

$$\det(A^2) = (\det(A))^2$$

- The determinant of a scalar multiple of  $A$  is the scalar raised to the  $n$ -th power times the determinant of  $A$ , where  $A$  is an  $n \times n$  matrix and  $c$  is a scalar:

$$\det(cA) = c^n \det(A)$$

12. The determinant of the transpose of  $A$  is equal to the determinant of  $A$ :

$$\det(A^T) = \det(A).$$

## 2.2 Rank and Trace

Rank of a matrix	Definition 2.2.1
<p>Given an <math>m \times n</math> matrix <math>A</math>, the rank of <math>A</math>, denoted <math>\text{rank}(A)</math>, is the number of linearly independent rows or columns; in particular, if <math>m = n</math>, <math>\text{rank}(A) = n</math> if and only if <math>\det(A) \neq 0</math>.</p> <p>Properties (without proof):</p> <ul style="list-style-type: none"> <li>• For a rectangular matrix <math>\text{rank}(A) \leq \min(n, m)</math>.</li> <li>• <math>\text{rank}(AB) \leq \min(\text{rank}(A), \text{rank}(B))</math>.</li> </ul>	
Trace of a matrix	Definition 2.2.2
<p>The trace of an <math>n \times n</math> matrix is defined as the sum of the elements along the main diagonal. Moreover:</p> <ul style="list-style-type: none"> <li>• <math>\text{trace}(ABC) = \text{trace}(BCA) = \text{trace}(CAB)</math>.</li> <li>• (Frobenius norm): <math>\ A\ _F = \sqrt{\sum_{i,j}  a_{i,j} ^2} = \sqrt{\text{trace}(A^H A)}</math> where <math>A^H</math> denotes the conjugate transpose of <math>A</math>.</li> </ul>	

## 2.3 Eigenvectors and Eigenvalues

Consider a matrix  $A$  and a vector  $x$ . The operation  $Ax$  produces a vector  $y$  at some direction. We are interested in vectors  $y$  which lie in the same direction as  $x$ .

In that case we have  $\mathbf{Ax} = \lambda \mathbf{Ix}$  with  $\lambda$  being a scalar. Because the LHS is matrix multiplication and the RHS is scalar multiplication, we generally add in the identity matrix  $I$  to help distinguish this:

$$\begin{aligned} \mathbf{Ax} &= \lambda \mathbf{Ix} \\ (\mathbf{A} - \lambda \mathbf{I})\mathbf{x} &= 0 \end{aligned}$$

When the above relationship holds,  $x$  is called an *eigenvector* and  $\lambda$  is called an *eigenvalue* of matrix  $A$ .

To solve this, if we know  $(\mathbf{A} - \lambda \mathbf{I})\mathbf{x} = 0$ , then  $\det((\mathbf{A} - \lambda \mathbf{I})\mathbf{x}) = 0$

If  $A$  is singular then  $\lambda = 0$  is an eigenvalue.

Why are we interested in the eigenvectors and eigenvalues of a matrix? Because they make the solution of complex problems simpler. For example, if we know the eigenvectors and eigenvalues of a linear mapping  $y = Ax$  we can characterize that mapping easily.

If  $A = \begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix}$ , eigenvalues:  $\lambda_1 = 4$ ,  $\lambda_2 = 2$ , eigenvectors:  $u_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ ,  $u_2 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$ . Then  $x = a_1 u_1 + a_2 u_2$  implies  $y = \lambda_1 a_1 u_1 + \lambda_2 a_2 u_2$ .

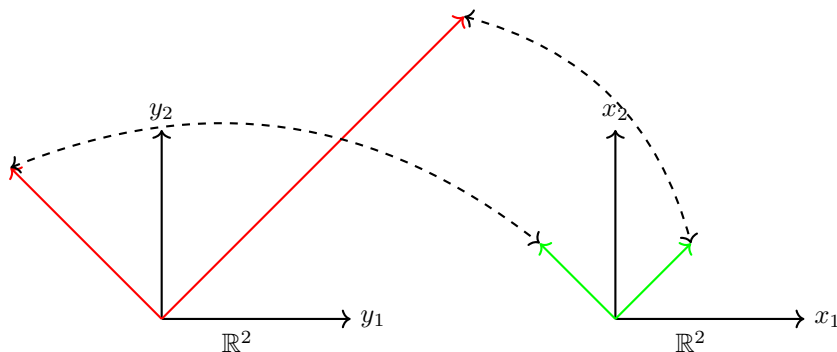


Figure 2.1: Characterised mapping

In electrical engineering, we have already seen a few examples of this, for example, in continuous-time, such as the Signals and Systems module on eigen-functions of an LTI system leading to the convolution formula in the Fourier domain, and the solutions of linear differential equations in Control Systems.

In the discrete-time finite dimensional case, eigenvalues and eigenvectors are very important for discrete-time convolution and for difference equations.

### 2.3.1 Eigenvectors and Eigenvalues of a permutation matrix

Consider permutation matrix  $\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ .

It will have eigenvector  $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$  with corresponding eigenvalue  $\lambda = 1$ , and eigenvector  $\begin{bmatrix} -1 \\ 1 \end{bmatrix}$  with corresponding eigenvalue  $\lambda = -1$ .

Note that  $n \times n$  matrices will have  $n$  eigenvalues, but it is not easy to find them. In this syllabus, we do not go into much detail about the properties of eigenvalues and eigenvectors of a permutation matrix, but we can say that since permutation matrices are orthogonal, the eigenvectors span an orthogonal basis.

#### Properties

- The sum of eigenvalues, the **trace** of the matrix, equals the sum of the diagonal elements of the matrix.
- The product of eigenvalues equals the determinant of the matrix.
- We could have worked out the second eigenvalue of  $\mathbf{A}$  by simply dividing the determinant by the first eigenvalue

### 2.3.2 Solving for Eigenvalues and Eigenvectors

#### Finding Eigenvalues

#### Definition 2.3.1

The eigenvalues of a matrix  $A$  are found by solving the characteristic equation:

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

where  $I$  is the identity matrix of the same size as  $A$ .

Once the eigenvalues  $\lambda_i$  are found, the corresponding eigenvectors  $v_i$  are found by solving the system of linear equations given by:

$$(A - \lambda_i I)v_i = 0$$

This system is solved for each eigenvalue  $\lambda_i$ . Note that there exists families of eigenvectors, not single eigenvectors.

### Example

For matrix  $A = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$ , we find eigenvalues by solving  $\det(A - \lambda I) = 0$ :

$$\begin{vmatrix} 2 - \lambda & 1 \\ 1 & 2 - \lambda \end{vmatrix} = (2 - \lambda)^2 - 1 = \lambda^2 - 4\lambda + 3 = 0$$

This gives  $\lambda_1 = 1$  and  $\lambda_2 = 3$ . The eigenvectors are found by substituting  $\lambda_i$  back into  $(A - \lambda_i I)v_i = 0$ .

### Example

Consider the matrix

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

The sum of the eigenvalues  $\lambda_1 + \lambda_2$  is equal to the trace of  $A$ , which is 6, and the product  $\lambda_1 \lambda_2$  is equal to the determinant of  $A$ , which is 9.

To find the eigenvalues, we calculate the determinant of  $A - \lambda I$ :

$$\det(A - \lambda I) = \det \begin{bmatrix} 3 - \lambda & 1 \\ 0 & 3 - \lambda \end{bmatrix} = (3 - \lambda)^2 = 0 \Rightarrow \lambda_{1,2} = 3.$$

**The eigenvalues of a triangular matrix are the values of the diagonal.**

For this particular case, we have:

$$\begin{bmatrix} 3 & 1 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 3x \\ 3y \end{bmatrix} \Rightarrow y = 0 \text{ and } x \text{ can be any number.}$$

### 2.3.3 Special Case: Eigenvalues with Identity Matrix Addition

Consider the matrix  $\mathbf{A} = \mathbf{B} + c\mathbf{I}$ .

Consider an eigenvector  $\mathbf{x}$  and eigenvalue  $\lambda$  of  $\mathbf{B}$ .

Then  $\mathbf{B}\mathbf{x} = \lambda\mathbf{x}$  and so:

$$\begin{aligned} \mathbf{A}\mathbf{x} &= (\mathbf{B} + c\mathbf{I})\mathbf{x} \\ &= \mathbf{B}\mathbf{x} + c\mathbf{I}\mathbf{x} \\ &= \mathbf{B}\mathbf{x} + c\mathbf{x} \\ &= \lambda\mathbf{x} + c\mathbf{x} \\ &= (\lambda + c)\mathbf{x} \end{aligned}$$

From this we can say  $\mathbf{A}$  has the same eigenvectors as  $\mathbf{B}$  with eigenvalues  $\lambda + c$ .

## 2.4 Diagonalisation of Matrices

- Suppose we have  $n$  linearly independent eigenvectors of a matrix  $A$ , denoted by  $\mathbf{x}_i$ .
- The corresponding eigenvalues are  $\lambda_i$ .
- These eigenvectors are arranged as columns in a matrix  $S$ , forming the so-called eigenvector matrix.

- By forming the product  $AS$ , where  $S = [\mathbf{x}_1 \mathbf{x}_2 \dots \mathbf{x}_n]$ , we effectively scale each eigenvector by its corresponding eigenvalue:

$$AS = A[\mathbf{x}_1 \mathbf{x}_2 \dots \mathbf{x}_n] = [\lambda_1 \mathbf{x}_1 \lambda_2 \mathbf{x}_2 \dots \lambda_n \mathbf{x}_n] = S\Lambda$$

where  $\Lambda$  is a diagonal matrix with eigenvalues  $\lambda_i$  on its diagonal and zeros elsewhere:

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n \end{bmatrix}$$

- This gives us the fundamental relationship  $AS = S\Lambda$ , which implies  $A = SAS^{-1}$  when  $S$  is invertible.
- The process of transforming  $A$  into  $\Lambda$  through the similarity transformation  $S$  is called *matrix diagonalisation*.
- Matrix diagonalisation is particularly important in Mathematics and Engineering because it simplifies many matrix computations and provides a clear framework for understanding linear transformations.

### 2.4.1 Matrix Diagonalisation and Commutativity

**Theorem:** Matrices  $A$  and  $B$  are simultaneously diagonalisable, sharing the same eigenvector matrix  $S$ , if and only if  $AB = BA$ .

*Sketch of the proof:* Given that  $A$  and  $B$  are diagonalisable, we can write  $A = S\Lambda_1 S^{-1}$  and  $B = S\Lambda_2 S^{-1}$ . Thus,

$$AB = S\Lambda_1 S^{-1} S\Lambda_2 S^{-1}$$

and

$$BA = S\Lambda_2 S^{-1} S\Lambda_1 S^{-1}$$

. Since  $\Lambda_1 \Lambda_2 = \Lambda_2 \Lambda_1$  (as they are diagonal matrices), it follows that  $AB = BA$ .

A matrix has  $n$  independent eigenvectors and therefore is diagonalisable if all the eigenvalues are different.

If  $Ax = \lambda x$  then  $A^2 x = A(Ax) = A(\lambda x) = \lambda(Ax) = \lambda^2 x$ . Therefore, the eigenvalues of  $A^2$  are  $\lambda^2$ . The eigenvectors of  $A^2$  remain the same since  $A^2 = SAS^{-1}SAS^{-1} = S\Lambda^2 S^{-1}$ . In general, for any positive integer  $k$ ,  $A^k = S\Lambda^k S^{-1}$ .

If there are repeated eigenvalues a matrix may, or may not have independent eigenvectors. For example, consider the identity matrix. Its eigenvectors are the row (or column) vectors. They are all independent, but the eigenvalues are all equal to 1.

#### A and B sharing Eigenvectors

#### Example Question 2.4.1

##### Question

If  $A$  and  $B$  share the same eigenvectors, what can we say about the eigenvalues of  $AB$ ?

##### Answer

Since  $A$  and  $B$  share the same eigenvectors,  $AB$  will also have the same eigenvectors. Let  $v$  be an eigenvector of  $A$  and  $B$ , then  $Av = \lambda v$  and  $Bv = \mu v$  for some eigenvalues  $\lambda$  and  $\mu$  of  $A$  and  $B$  respectively. Thus:

$$ABv = A(\mu v) = \mu Av = \mu \lambda v$$

Showing us that  $v$  is also an eigenvector of  $AB$  with corresponding eigenvalue  $\mu\lambda$ . But this does not tell us that all eigenvalues of  $AB$  are necessarily products of eigenvalues of  $A$  and  $B$ .  $A$  and  $B$  must be diagonalisable and  $A, B$  are commutative so  $AB = BA$ .  $A = PDP^{-1}$  and  $B = PQP^{-1}$ , where  $D$  and  $Q$  are diagonal matrices with eigenvalues of  $A$  and  $B$  respectively, and  $P$  is the matrix of shared eigenvectors. Then  $AB = PDQP^{-1}$ , and the eigenvectors form a basis, so diagonal entries of  $DQ$  will be eigenvalues of  $AB$  since the determinant between the two are preserved.

**Question**

If  $A = S\Lambda S^{-1}$  is square and invertible, can you compute  $A^{-1}$ ?

**Answer**

If  $A$  is invertible, then  $A^{-1} = S\Lambda^{-1}S^{-1}$ , where  $\Lambda^{-1}$  is the diagonal matrix with the reciprocal of  $A$ 's eigenvalues on the diagonal.

## 2.5 Difference Equations

### Fibonacci Sequence Analysis

The Fibonacci sequence (0,1,1,2,3,5,8,13...) is a series of numbers where each number is the sum of the two preceding ones, typically starting with 0 and 1. Mathematically, it can be expressed as:

$$F_{k+2} = F_{k+1} + F_k \quad (2.1)$$

where  $F_k$  represents the  $k$ -th term of the Fibonacci sequence.

To solve this using linear algebra, we can define the state vector  $u_k$ :

$$u_k = \begin{bmatrix} F_{k+1} \\ F_k \end{bmatrix} \quad (2.2)$$

By employing the recursive definition of the Fibonacci sequence, we obtain the matrix form:

$$u_{k+1} = Au_k = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} u_k \quad (2.3)$$

Given the initial state vector  $u_0$  that we know of, the state at any point  $k$  is given by:

$$u_k = A^k u_0 \quad (2.4)$$

If  $A$  can be diagonalized as  $A = S\Lambda S^{-1}$ , where  $\Lambda$  is a diagonal matrix of eigenvalues and  $S$  is a matrix of corresponding eigenvectors, then the computation simplifies to:

$$u_k = S\Lambda^k S^{-1} u_0 \quad (2.5)$$

Thus, knowing the eigenvectors and eigenvalues of  $A$  can greatly simplify the computation of terms in the Fibonacci sequence.

### System of Difference Equations

Consider a system described by a set of difference equations:

$$y_1[n+1] = y_1[n] - 1.5y_2[n] \quad (2.6)$$

$$y_2[n+1] = 0.5y_1[n] + y_2[n] \quad (2.7)$$

This system can be written in matrix form using the state vector  $u_n$ :

$$u_n = \begin{bmatrix} y_1[n] \\ y_2[n] \end{bmatrix} \quad (2.8)$$

Thus, the next state  $u_{n+1}$  is given by:

$$u_{n+1} = Au_n = \begin{bmatrix} 1 & -1.5 \\ 0.5 & 1 \end{bmatrix} u_n \quad (2.9)$$

Given  $u_0$ , the state  $u_n$  is then:

$$u_n = A^n u_0 \quad (2.10)$$

As with the Fibonacci sequence, if  $A$  can be diagonalized, the solution becomes more tractable:

$$u_n = S\Lambda^n S^{-1} u_0 \quad (2.11)$$

Hence, eigenvalues and eigenvectors are key to simplifying the solution of the system of difference equations, providing a powerful method to analyze and solve linear discrete-time systems in signals and systems.

## 2.6 Markov Processes

Markov processes are used to model systems that undergo transitions from one state to another on a state space. A key feature of a Markov process is that the future state depends only on the current state and not on the sequence of events that preceded it.

Consider a Markov process example inspired by Strang's book, modelling population migration between California and the outside:

“Assume that each year 1/10 of the people outside California move in and 2/10 of the people inside California move out.”

Let  $y_0$  be the initial population outside California and  $z_0$  the initial population inside California. The state vector and the transition matrix  $A$  are given by:

$$\begin{bmatrix} y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} 0.9 & 0.2 \\ 0.1 & 0.8 \end{bmatrix} \begin{bmatrix} y_0 \\ z_0 \end{bmatrix} = A \begin{bmatrix} y_0 \\ z_0 \end{bmatrix} \quad (2.12)$$

The eigenvalues and eigenvectors of  $A$  are crucial for understanding the long-term behavior of the Markov process. In this case, the eigenvalues are  $\lambda_1 = 1$  and  $\lambda_2 = 0.7$ , with corresponding eigenvectors:

$$\mu_1 = \begin{bmatrix} 2/3 \\ 1/3 \end{bmatrix}, \quad \mu_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \quad (2.13)$$

We can then write the state at instant  $k$  as:

$$\begin{bmatrix} y_k \\ z_k \end{bmatrix} = A^k \begin{bmatrix} y_0 \\ z_0 \end{bmatrix} \quad (2.14)$$

For a large  $k$ , the contribution from the second eigenvalue becomes negligible, leading to a steady-state solution where the population distribution does not change from one year to the next:

$$\begin{bmatrix} y_\infty \\ z_\infty \end{bmatrix} = (y_0 + z_0) \begin{bmatrix} 2/3 \\ 1/3 \end{bmatrix} \quad (2.15)$$

### Properties of Transition Matrices

From this example, we can deduce some general properties of Markov or transition matrices:

- All entries are non-negative.
- Each column of the matrix adds up to one, ensuring the total probability is conserved.
- The eigenvalue  $\lambda_1 = 1$  corresponds to the steady state.
- All other eigenvalues  $\lambda_i$  satisfy  $|\lambda_i| \leq 1$ .

Additionally, it can be shown that  $\lambda_1 = 1$  is always an eigenvalue by noting that each column of  $A - I$  sums to zero, indicating that  $A - I$  is singular and thus  $\lambda_1 = 1$  is an eigenvalue. The corresponding eigenvector represents the steady-state distribution since  $A\mu_1 = \mu_1$ . This implies that in the long run, the system will reach a state where the population distribution remains constant if no external changes are made to the system.

### Computing the State at Instant $k$

Given the eigendecomposition of  $A$  as  $A = S\Lambda S^{-1}$ , where  $S$  is the matrix of eigenvectors and  $\Lambda$  is the diagonal matrix of eigenvalues, the state at any instant  $k$  can be computed as:

$$\begin{bmatrix} y_k \\ z_k \end{bmatrix} = S\Lambda^k S^{-1} \begin{bmatrix} y_0 \\ z_0 \end{bmatrix} \quad (2.16)$$

For our specific matrix  $A$ , the computation yields:

$$\begin{bmatrix} y_k \\ z_k \end{bmatrix} = \begin{bmatrix} 2/3 & 1/3 \\ 1/3 & -1/3 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & (0.7)^k \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -2 \end{bmatrix} \begin{bmatrix} y_0 \\ z_0 \end{bmatrix} \quad (2.17)$$

## Steady-State Solution

The steady-state solution, which occurs as  $k$  approaches infinity, is given by the eigenvector corresponding to  $\lambda_1 = 1$ :

$$\begin{bmatrix} y_\infty \\ z_\infty \end{bmatrix} = (y_0 + z_0) \begin{bmatrix} 2/3 \\ 1/3 \end{bmatrix} \quad (2.18)$$

This indicates that regardless of the initial state, the system will evolve into a state where the ratios of the populations inside and outside California are 2 : 1, respectively.

## General Observations

The model demonstrates a fundamental principle of Markov processes: despite potentially complex dynamics, the long-term behaviour is determined by the eigenvalues and eigenvectors of the transition matrix, with the largest eigenvalue indicating the steady state. This simplifies the analysis of systems described by Markov processes, particularly in the fields of signals and systems, stochastic processes, and population dynamics.

## 2.7 Diagonalisation of Circulant Matrices

Diagonalization of circulant matrices is closely tied to the concept of convolution in signal processing. A circulant matrix  $A$  can model the circular convolution operation between an input vector  $\mathbf{x}$  and a filter  $\mathbf{h}$ , resulting in an output vector  $\mathbf{y}$  such that  $\mathbf{y} = A\mathbf{x}$ .

### Discrete Fourier Transform (DFT)

The Discrete Fourier Transform (DFT) plays a pivotal role in the analysis of circulant matrices. For a DFT matrix of size  $n \times n$ , the element in the  $i$ -th row and  $k$ -th column is given by  $w_n^{ik}$ , where  $w_n = e^{-j\frac{2\pi}{n}}$ . The DFT matrix  $W$  has the following form:

$$W = \frac{1}{\sqrt{n}} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & w_n & w_n^2 & \cdots & w_n^{n-1} \\ 1 & w_n^2 & w_n^4 & \cdots & w_n^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & w_n^{n-1} & w_n^{2(n-1)} & \cdots & w_n^{(n-1)(n-1)} \end{bmatrix}$$

### Eigenvectors and Eigenvalues

The eigenvectors of a circulant matrix are precisely the columns of the DFT matrix  $W$ . These eigenvectors, denoted as  $\mathbf{u}_i$ , can be expressed as:

$$\mathbf{u}_i = \frac{1}{\sqrt{n}} \begin{bmatrix} 1 \\ w_n^i \\ w_n^{2i} \\ \vdots \\ w_n^{(n-1)i} \end{bmatrix}$$

The eigenvalues of a circulant matrix are the DFT of the first row of the matrix, which corresponds to the filter  $\mathbf{h}$ . This direct correspondence facilitates the use of DFT in signal processing to perform efficient convolution operations.

### Diagonalization by DFT

The DFT matrix  $W$  has the property of diagonalizing a circulant matrix  $A$ . This means that  $A$  can be represented as  $A = W\Lambda W^{-1}$ , where  $\Lambda$  is a diagonal matrix containing the eigenvalues of  $A$ . As a result, the DFT simplifies the multiplication of a circulant matrix with a vector, transforming it into a multiplication of the diagonal matrix  $\Lambda$  with the DFT of the vector.

### Implications in Signal Processing

In signal processing, this diagonalization property is exploited for efficient computation of convolutions, particularly using algorithms such as the Fast Fourier Transform (FFT), which is a computationally efficient implementation of the DFT. Thus, the DFT forms the basis for many signal processing techniques, enabling the analysis and manipulation of signals in the frequency domain.



## 2.8 Complex, Symmetric, Hermitian Matrices

Symmetric matrices are a fundamental concept in linear algebra with important implications in various fields, including signal processing and systems theory.

### Definition and Properties

- A real matrix  $A$  is said to be symmetric if  $A = A^T$ , where  $A^T$  denotes the transpose of  $A$ .
- For complex matrices, symmetry is defined by the condition  $A^* = A$ , where  $A^*$  denotes the conjugate transpose of  $A$ . Such matrices are also known as *Hermitian matrices*.
- The notation  $A^H$  is also commonly used to denote the conjugate transpose, so for Hermitian matrices,  $A^H = A$ .
- Symmetric matrices are prevalent in numerous applications, notably in statistics as covariance matrices.
- It is an established fact that the eigenvalues of a symmetric matrix are real numbers. This will be demonstrated through a proof.
- Additionally, eigenvectors of a symmetric matrix corresponding to distinct eigenvalues are orthogonal. The proof of this statement is provided in a subsequent section.
- For symmetric matrices with orthonormal eigenvectors, we can represent  $A$  as  $A = Q\Lambda Q^T$ , where  $Q$  is the matrix of eigenvectors, and  $\Lambda$  is the diagonal matrix of eigenvalues.

### Eigenvalues and Eigenvectors

#### Proof of Complex Conjugate Pairs for Eigenvalues

#### Example Question 2.8.1

##### Question

Prove that the eigenvalues of a symmetric matrix occur in complex conjugate pairs.

##### Solution

Consider the eigenvalue equation  $Ax = \lambda x$ .

- Taking the complex conjugate of both sides of the equation yields  $(Ax)^* = (\lambda x)^*$  which simplifies to  $A^*x^* = \lambda^*x^*$ .
- If  $A$  is a real symmetric matrix, then  $A^* = A$ , and hence  $\lambda^*x^*$  must also be an eigenvector-eigenvalue pair, i.e., if  $\lambda$  is an eigenvalue of  $A$  with corresponding eigenvector  $x$ , then  $\lambda^*$  is also an eigenvalue of  $A$  with corresponding eigenvector  $x^*$ .

#### Proof that Symmetric Matrices' Eigenvalues are Real

#### Example Question 2.8.2

##### Question

Prove that the eigenvalues of a symmetric matrix are real.

##### Solution

We have previously shown that for a real matrix  $A$ , the complex conjugate of an eigenvalue  $\lambda$  is also an eigenvalue. Let's consider the eigenvalue equation  $Ax = \lambda x$ .

1. Taking the transpose of both sides of the equation yields  $x^T A^T = \lambda x^T$ . Since  $A$  is symmetric,  $A^T = A$ , so this becomes  $x^T A = \lambda x^T$ .
2. Multiplying both sides from the right by  $x$ , we get  $x^T A x = \lambda x^T x$ .
3. Now, we take the eigenvalue equation  $Ax = \lambda x$  and multiply both sides from the left with  $x^T$  to obtain  $x^T A x = \lambda x^T x$ .
4. Observing the above, since  $x^T x$  is a scalar and  $x^T x \neq 0$ , it implies that  $\lambda$  must be equal to its complex conjugate  $\lambda^*$ , which means  $\lambda$  is real.

### Proof that Eigenvectors of a Symmetric Matrix with Unique Eigenvalues are always Orthogonal Example Question 2.8.3

#### Question

Prove that the eigenvectors of a symmetric matrix corresponding to different eigenvalues are always orthogonal.

#### Solution

Let  $Ax = \lambda_1 x$  and  $Ay = \lambda_2 y$  be two eigenvectors of  $A$  corresponding to distinct eigenvalues  $\lambda_1 \neq \lambda_2$ .

1. Taking the dot product of  $Ax$  with  $y$  yields  $(Ax)^T y = \lambda_1 x^T y$ .
2. Since  $A$  is symmetric,  $(Ax)^T y = x^T A^T y = x^T Ay = x^T \lambda_2 y$ .
3. These conditions imply  $\lambda_1 x^T y = \lambda_2 x^T y$  and since  $\lambda_1 \neq \lambda_2$ , it follows that  $x^T y = 0$ .
4. Thus, the eigenvectors  $x$  and  $y$  are orthogonal.

### A Hermitian Matrix always has Real Eigenvectors

### Definition 2.8.1

- We aim to determine which complex matrices have real eigenvalues and orthogonal eigenvectors.
- Consider the eigenvalue equation  $Ax = \lambda x$  with  $A$  possibly complex.
- Taking the complex conjugate on both sides yields  $(Ax)^* = (\lambda x)^*$  which implies  $A^* x^* = \lambda^* x^*$ .
- By transposing both sides, we obtain  $x^{*T} A^{*T} = \lambda^* x^T$ .
- Multiplying both sides from the right by  $x$ , we get  $x^* A^* x = \lambda^* x^T x$ .
- Starting again with  $Ax = \lambda x$  and multiplying both sides from the left by  $x^*$ , we have  $x^* Ax = \lambda x^* x$ .

From the above steps, we deduce that if  $A^{*T} = A$ , then  $\lambda x^* x = \lambda^* x^* x$  and since  $x^* x \neq 0$ , it follows that  $\lambda = \lambda^*$ .

### Proof of Orthogonal Eigenvectors for unique Eigenvectors in Hermitian Matrices Example Question 2.8.4

#### Question:

Prove that the eigenvectors of a complex symmetric matrix (Hermitian matrix) which correspond to different eigenvalues are always perpendicular.

#### Solution:

Suppose  $Ax = \lambda_1 x$  and  $Ay = \lambda_2 y$ , where  $\lambda_1 \neq \lambda_2$ . We have

$$(\lambda_1 x)^H y = x^H \lambda_1 y = (Ax)^H y = x^H Ay = x^H \lambda_2 y$$

The conditions  $x^H \lambda_1 y = x^H \lambda_2 y$  and  $\lambda_1 \neq \lambda_2$ , give  $x^H y = 0$ , so these eigenvalues  $x$  and  $y$  are perpendicular.

## 2.8.1 Spectral Theorem

We have demonstrated that:

- The eigenvalues of a symmetric matrix, whether real or complex, are real.
- The eigenvectors of a symmetric matrix, whether real or complex, that correspond to different eigenvalues are orthogonal.

We conclude this part with the following statement (without proof):

- **Spectral theorem:** Every real symmetric matrix can be diagonalised by an orthogonal matrix, and every Hermitian matrix can be diagonalised by a unitary matrix:
  - For the real case:  $A = Q\Lambda Q^{-1} = Q\Lambda Q^T$
  - For the complex case:  $A = Q\Lambda Q^{-1} = Q\Lambda Q^H$

## 2.8.2 Positive Definite Hermitian Matrices

Positive Definite Hermitian Matrices	Definition 2.8.2
<p>Hermitian matrix <math>\mathbf{A}</math> is positive definite if and only if:</p> <ul style="list-style-type: none"> <li>• <math>\mathbf{x}^H \mathbf{A} \mathbf{x} &gt; 0</math> and <math>\mathbf{x} \neq 0</math></li> <li>• All eigenvalues of <math>\mathbf{A}</math> satisfy <math>\lambda_i &gt; 0</math></li> <li>• All pivots without row exchanges satisfy <math>d_i &gt; 0</math></li> </ul>	

Positive Definite Test for Identifying Stationary Points	Example Question 2.8.5
<p>Consider the function <math>f(x_1, x_2) = a_{11}x_1^2 + 2a_{12}x_1x_2 + a_{22}x_2^2</math>.</p> <ul style="list-style-type: none"> <li>• It is evident that <math>f(0, 0) = 0</math> and at the point <math>(0, 0)</math>, we also have <math>\frac{\partial f}{\partial x_1} = \frac{\partial f}{\partial x_2} = 0</math>.</li> <li>• The question is: Is <math>(0, 0)</math> a minimum, a maximum, or a saddle point?</li> <li>• If the matrix <math>A</math> defined as</li> </ul> $A = \begin{bmatrix} a_{11} & a_{12} \\ a_{12} & a_{22} \end{bmatrix}$ <p>is positive definite, then <math>(0, 0)</math> is a minimum. If <math>A</math> is negative definite, then it is a maximum. If <math>A</math> is indefinite, then <math>(0, 0)</math> is a saddle point.</p> <p>In general, for a function in several variables of the form <math>f(\mathbf{x}) = \sum_{i=1}^n \sum_{j=1}^n a_{ij}x_i x_j</math>, it can be written in a compact matrix form as <math>\mathbf{x}^T \mathbf{A} \mathbf{x}</math>. By testing whether <math>A</math> is positive definite, we can ascertain if the point at <math>\mathbf{x} = \mathbf{0}</math> is a global minimum.</p>	

## 2.9 Graph Adjacency Matrix

Graphs are mathematical structures used to model pairwise relations between objects. A graph is made up of *nodes* (or *vertices*) and *edges* that connect pairs of nodes. Graphs can be used to represent various systems in the real world, such as:

- Social networks: where people are nodes and friendships are links.
- Correlated variables: with variables as nodes and correlations as links.
- Web pages and hyperlinks: where each web page is a node and hyperlinks are edges.

### Formal Definition

A graph  $G = (V, E)$  is defined by:

- A set  $V$  of nodes (vertices):  $V = \{1, 2, 3, 4, 5, 6\}$ .
- A set  $E$  of pairs of nodes denoting edges:  $E = \{\{1, 2\}, \{1, 5\}, \{2, 5\}, \{2, 3\}, \{3, 4\}, \{4, 5\}, \{4, 6\}\}$ .

Graphs can be *directed* if edges have directions, or *undirected* otherwise. They are *weighted* if we associate a value (weight) with each edge, and *unweighted* if not.

### Adjacency Matrix

The adjacency matrix  $A$  of a graph is defined by:

$$a_{i,j} = \begin{cases} w_{i,j} & \text{if } i \text{ is connected to } j, \\ 0 & \text{otherwise.} \end{cases}$$

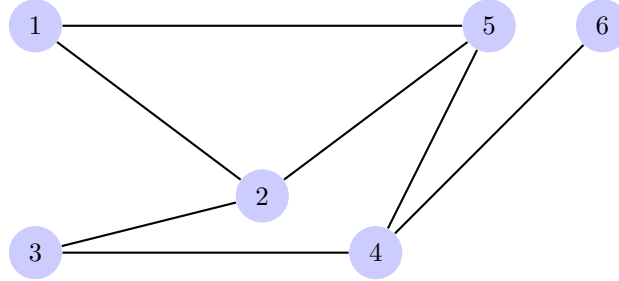
For an unweighted and undirected graph, the adjacency matrix is symmetric, and its entries are 1 if there is an edge between the corresponding nodes and 0 otherwise.

## Degree of a Node

The degree  $d_i$  of a node is the number of nodes to which it is adjacent. For example, the degree of node 4,  $d_4$ , is 3. In an undirected and unweighted graph, the degree of node  $i$  is obtained by summing across the  $i$ -th row or column of  $A$ .

## Example Graph

The following diagram represents the undirected graph  $G$ :



## Adjacency Matrix of the Example Graph

The adjacency matrix  $A$  for the undirected graph  $G$  is:

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

## 2.10 Pagerank Algorithm

The PageRank algorithm measures the importance of each node within a graph. It gives the idea that connections to high-importance nodes contribute more to the importance of the node in question. This importance is quantified through a vector  $\mathbf{u}$ , where each entry  $u_i$  corresponds to the centrality (importance) of node  $i$ .

### Centrality and Eigenvector Problem

The centrality of a node is not just a function of its degree  $d_i$ , but also the centralities of the nodes pointing to it. The centrality  $u_i$  for node  $i$  can be expressed as:

$$u_i = \sum_{j=1}^n a_{ij} u_j$$

This is the sum of the connection weights to a nodes other neighbours, multiplied by the importance of the neighbour, which can be performed by matrix operation  $A\mathbf{u}$ . When optimising for the weights, we expect to reach convergence when we iteratively solve this eigenvector problem:

$$A\mathbf{u} = \lambda\mathbf{u}$$

where  $A$  is the adjacency matrix of the graph, and  $\lambda$  is a constant. This is an eigenvalue problem, because given an adjacency matrix  $\mathbf{A}$  and a vector  $\mathbf{u}$  of correct importance scores for the given  $\mathbf{A}$ , their product (which is an act of calculating importance scores) should still net the same  $\mathbf{u}$ . This makes  $\mathbf{u}$  the principal eigenvector that gives the centrality scores. We want the entries of  $\mathbf{u}$  to be non-negative, so we want the largest eigenvalue of  $A$  (proof for this is not shown).

### Power Method for PageRank

To find the principal eigenvector, the Power Method is utilised:

1. Assume matrix  $\mathbf{A}$  with a full set of eigenvectors ( $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n$ ) and no multiple eigenvalues.

2. Start with a non-negative initial guess vector  $\mathbf{x}_0$ .
3. Iterate  $\mathbf{x}_k = \frac{A\mathbf{x}_{k-1}}{\|A\mathbf{x}_{k-1}\|}$ , and normalise  $\mathbf{x}_k$  at each step to prevent divergence. It is not specified in the notes which norm to use (it is assumed to be the L2 norm following the notation of the module), some other practices take the infinity norm which is the largest component in the vector.

As  $k$  approaches infinity,  $\mathbf{x}_k$  converges to the principal eigenvector  $\mathbf{u}$ , corresponding to the largest eigenvalue, which is used as the PageRank score.

This is because

$$\mathbf{x}_k = A\mathbf{x}_{k-1} = A^k\mathbf{x}_0 = U\Lambda^k U^{-1}\mathbf{x}_0 = c_1\lambda_1^k u_1 + c_2\lambda_2^k u_2 + \cdots + c_n\lambda_n^k u_n$$

Expanding the equation:

$$\mathbf{x}_k = U\Lambda^k U^{-1}\mathbf{x}_0 = \begin{bmatrix} | & | & & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_n \\ | & | & & | \end{bmatrix} \begin{bmatrix} \lambda_1^k & 0 & \cdots & 0 \\ 0 & \lambda_2^k & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n^k \end{bmatrix} \begin{bmatrix} | & | & & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_n \\ | & | & & | \end{bmatrix}^{-1} \mathbf{x}_0$$

The eigenvalues  $\lambda_i$  are typically ordered such that  $|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_n|$ , and the coefficients  $c_i$  are the components of  $\mathbf{x}_0$  in the basis of eigenvectors  $\mathbf{u}_i$ . As  $k$  increases,  $\lambda_1^k$  dominates the sum, and  $\mathbf{x}_k$  converges to a scalar multiple of the principal eigenvector  $\mathbf{u}_1$ :

$$\mathbf{x}_k \approx c_1\lambda_1^k \mathbf{u}_1$$

where  $c_1$  is the component of the initial guess  $\mathbf{x}_0$  in the direction of  $\mathbf{u}_1$ .

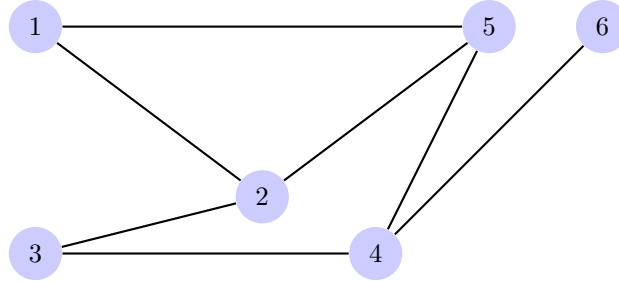
Note that this algorithm is standard, not too performant, but works for manual calculation.

## Example Application of PageRank

Considering our graph  $G$  with nodes  $\{1, 2, 3, 4, 5, 6\}$  and edges  $\{(1, 2), (1, 5), (2, 5), (4, 5), (2, 3), (3, 4), (4, 6)\}$ , let us calculate the PageRank scores.

### Graph Representation

The graph is represented as:



### Adjacency Matrix and PageRank Scores

We start with initial guess  $\mathbf{x}_0$

$$\mathbf{x}_0 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \text{ and } A = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

$$\mathbf{x}_1 = A\mathbf{x}_0 = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \\ 2 \\ 3 \\ 3 \\ 1 \end{bmatrix} \xrightarrow{\text{normalise}} \begin{bmatrix} 0.3333 \\ 0.5000 \\ 0.3333 \\ 0.5000 \\ 0.5000 \\ 0.1667 \end{bmatrix}$$

$$\begin{aligned}
\mathbf{x}_2 = \mathbf{A}\mathbf{x}_1 &= \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0.333 \\ 0.500 \\ 0.333 \\ 0.500 \\ 0.500 \\ 0.167 \end{bmatrix} \xrightarrow{\text{normalise}} \begin{bmatrix} 0.396 \\ 0.462 \\ 0.396 \\ 0.396 \\ 0.528 \\ 0.198 \end{bmatrix} \\
\mathbf{x}_3 = \mathbf{A}\mathbf{x}_2 &= \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0.396 \\ 0.462 \\ 0.396 \\ 0.396 \\ 0.528 \\ 0.198 \end{bmatrix} \xrightarrow{\text{normalise}} \begin{bmatrix} 0.390 \\ 0.520 \\ 0.338 \\ 0.442 \\ 0.494 \\ 0.156 \end{bmatrix} \\
&\vdots \\
&\mathbf{x}_{50} = \begin{bmatrix} 0.401 \\ 0.502 \\ 0.358 \\ 0.408 \\ 0.516 \\ 0.160 \end{bmatrix}
\end{aligned}$$

It does follow after many iterations that node 2 and node 5 are the two most central nodes in this network, and it is notable that they both are linked together and have the highest number of connections, 3. Although node 4 has 3 connections as well, it is connected to nodes of lesser importance, resulting in a lower rank.

# Chapter 3

## Solving $\mathbf{Ax} = \mathbf{b}$

### 3.1 Interpretations

We aim to solve  $\mathbf{Ax} = \mathbf{b}$  where  $\mathbf{b} \in \mathbb{C}^m$  and  $\mathbf{A} \in \mathbb{C}^{m \times n}$ , and we want to determine if solution  $\mathbf{x}$  exists and whether it is unique.

Otherwise, if there is no solution, we pick the best approximation. We are also interested when  $m \neq n$ , when there is no solutions, or if they are not unique.

Estimating  $\mathbf{x}$  from  $\mathbf{Ax} = \mathbf{b}$  is an inverse problem. An example in engineering: when the equations in  $\mathbf{A}$  models a distortion filter operation on some data, and we want to want restore  $\mathbf{x}$ . An example of distortion could be motion blur in a photo, or room echo in a sound recording.



### Geometric Interpretation

In the context of linear algebra, the equation  $Ax = b$  can be interpreted in both pure mathematical and geometrical terms as a system of linear equations.

- When  $m = n$ , we have an equal number of equations and unknowns, which constitutes a square system. For instance, consider the system

$$\begin{pmatrix} 1 & 2 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 4 \\ 1 \end{pmatrix},$$

where each equation represents a line in a two-dimensional space. The intersection of two non-parallel lines signifies a unique solution.

- If the lines represented by the equations are parallel, the system may either have no solution or infinitely many solutions (the lines coincide). The *rank* of matrix  $A$  is a critical determinant in identifying the uniqueness of the solution.
- In the case where  $m > n$ , implying more equations than unknowns, the system is referred to as ‘tall’. Generally, such systems do not have a unique solution, as the equations may intersect at multiple points unless several of them overlap.
- Conversely, when  $m < n$ , meaning there are more unknowns than equations, the system is described as ‘fat’. An example of this is when  $m = 2$  and  $n = 3$ ; we are dealing with two planes in a three-dimensional space. Unless the planes are parallel, they intersect along a line, leading to an infinite number of solutions. Thus, the solution, if it exists, is not unique.

## Vector Space Interpretation

Consider the matrix equation  $\mathbf{Ax} = \mathbf{b}$ . This can be interpreted within the framework of vector spaces as follows:

- The matrix  $\mathbf{A}$  can be seen as a linear transformation that maps vectors from  $\mathbb{C}^n$  to  $\mathbb{C}^m$ .
- The operation  $\mathbf{Ax}$  corresponds to forming linear combinations of the columns of  $\mathbf{A}$ , where the vector  $\mathbf{x}$  provides the combination coefficients.
- A solution to  $\mathbf{Ax} = \mathbf{b}$  is feasible if and only if the vector  $\mathbf{b}$  resides within the range (or column space) of  $\mathbf{A}$ .
- The range space of  $\mathbf{A}$ , denoted as  $\mathcal{R}(\mathbf{A})$ , contains all possible vectors  $\mathbf{b}$  that can be reached by the linear transformation  $\mathbf{A}$ . If  $\mathcal{R}(\mathbf{A}) = \mathbb{C}^m$ , meaning  $\mathbf{A}$  has full row rank, then any vector in  $\mathbb{C}^m$  can be obtained from  $\mathbf{A}$ .
- The existence of a solution is indicated by  $\mathbf{b}$  being in the range space of  $\mathbf{A}$ , while the null space of  $\mathbf{A}$  determines the uniqueness. Specifically, if the null space of  $\mathbf{A}$  contains only the zero vector (is trivial), the solution to  $\mathbf{Ax} = \mathbf{b}$  is unique.
- Conversely, if the null space of  $\mathbf{A}$  is non-trivial, possessing vectors other than the zero vector, multiple solutions to  $\mathbf{Ax} = \mathbf{b}$  may exist.

## 3.2 Existence and Uniqueness

The previous section is summarised by these theorems:

Existence	Theorem 3.2.1
<p>Let <math>\mathbf{A} \in \mathbb{C}^{m \times n}</math>.</p> <ol style="list-style-type: none"> <li>1. For each <math>\mathbf{b} \in \mathbb{C}^m</math> there exists at least one solution <math>\mathbf{Ax} = \mathbf{b}</math></li> <li>2. The range space of <math>\mathbf{A}</math> is full: <math>\mathcal{R}(\mathbf{A}) = \mathbb{C}^m</math></li> <li>3. <math>\text{rank}(\mathbf{A}) = m</math></li> <li>4. <math>\mathbf{A}</math> is full row rank, so its rows are linearly independent. These conditions imply <math>m \leq n</math>, a fat or square matrix.</li> </ol>	
Uniqueness	Theorem 3.2.2
<p>Let <math>\mathbf{A} \in \mathbb{C}^{m \times n}</math>.</p> <ol style="list-style-type: none"> <li>1. If there exists a solution to <math>\mathbf{Ax} = \mathbf{b}</math>, it is unique</li> <li>2. The null space of <math>\mathbf{A}</math> is trivial (containing nothing but the zero vector)</li> <li>3. <math>\text{rank}(\mathbf{A}) = n</math></li> <li>4. <math>\mathbf{A}</math> is full column rank, so its columns are linearly independent.</li> </ol> <p>These conditions imply <math>m \geq n</math>, a tall or square matrix.</p>	
Existence and Uniqueness	Theorem 3.2.3
<p>Let <math>\mathbf{A} \in \mathbb{C}^{m \times n}</math>.</p> <ol style="list-style-type: none"> <li>1. For each <math>\mathbf{b} \in \mathbb{C}^m</math> there exists at least one solution <math>\mathbf{Ax} = \mathbf{b}</math> and it is unique</li> <li>2. The range space of <math>\mathbf{A}</math> is <math>m</math>: <math>\mathcal{R}(\mathbf{A}) = \mathbb{C}^m</math></li> <li>3. <math>\text{rank}(\mathbf{A}) = m = n</math> and <math>\mathbf{A}</math>'s null space is trivial.</li> <li>4. <math>\mathbf{A}</math> is square and nonsingular, so it has a determinant <math>\neq 0</math></li> </ol>	

1. Consider the matrix

$$A = \begin{pmatrix} 4 & 8 & 6 \\ 0 & 4 & 7 \end{pmatrix}.$$



The rows are linearly independent, so for every vector  $\mathbf{b}$  in  $\mathbb{C}^2$ , there is at least one solution to  $A\mathbf{x} = \mathbf{b}$ .

2. The matrix

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

has three rows but only rank 2. The conditions for the existence theorem are not satisfied, and there exist vectors  $\mathbf{b}$  for which  $A\mathbf{x} = \mathbf{b}$  has no solution. For example,  $\mathbf{b} = (0 \ 0 \ 1)^T$ .

3. The matrix

$$A = \begin{pmatrix} 6 & 5 & 6 & 3 \\ 2 & 3 & 0 & 1 \\ 4 & 4 & 3 & 2 \end{pmatrix}$$

has more columns than rows and is not full row rank. Thus, the conditions for the existence theorem are not met, and a solution to  $A\mathbf{x} = \mathbf{b}$  does not always exist. Remember, having 'more unknowns than equations' or being 'underdetermined' does not guarantee the existence of a solution; having 'full row rank' does.

4. Given the matrix

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

the columns are linearly independent. Therefore, for every vector  $\mathbf{b}$  in  $\mathbb{C}^2$ , there exists a unique solution to  $A\mathbf{x} = \mathbf{b}$ . For example, if  $\mathbf{b} = \begin{pmatrix} 4 \\ 6 \end{pmatrix}$ , the unique solution is  $\mathbf{x} = \begin{pmatrix} 2 \\ 4 \end{pmatrix}$ . However, for  $\mathbf{b} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ , no solution exists.

5. For the matrix

$$A = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 2 \end{pmatrix},$$

the columns are linearly dependent, with the null space being the span of  $\begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix}$ . Thus, neither uniqueness nor existence of the solution is guaranteed.

6. Consider the matrix

$$A = \begin{pmatrix} 2 & 5 & 3 \\ 3 & 5 & 2 \\ 3 & 7 & 4 \\ 7 & 8 & 1 \end{pmatrix}.$$

The columns are linearly dependent, which means that even though the system is overdetermined, uniqueness of the solution cannot be assured.

Our goal is to develop a single method that either finds  $\mathbf{x}$  as the solution to  $A\mathbf{x} = \mathbf{b}$  if it exists, or find the best approximation.

### 3.3 Four Fundamental Subspaces

Given a linear mapping  $A : \mathbb{C}^n \rightarrow \mathbb{C}^m$  represented by the matrix  $A \in \mathbb{C}^{m \times n}$ , we consider two important concepts: the range (or column space) of  $A$ , denoted as  $\mathcal{R}(A)$ , and the null space (or kernel) of  $A$ , denoted as  $\mathcal{N}(A)$ .

Exploring further, let us examine the adjoint of  $A$ , denoted as  $A^H$ , which maps vectors from  $\mathbb{C}^m$  to  $\mathbb{C}^n$ . It is crucial to understand that  $A^H$  is not necessarily the inverse of  $A$  but rather the conjugate transpose of  $A$ .

The range of  $A^H$ ,  $\mathcal{R}(A^H)$ , and the null space of  $A^H$ ,  $\mathcal{N}(A^H)$ , can be defined analogously. These subspaces interact with each other in the following ways, which we state without proof:

- $\mathbb{C}^m = \mathcal{R}(A) \oplus \mathcal{N}(A^H)$ , where  $\oplus$  denotes the direct sum, and  $\mathcal{R}^\perp(A) = \mathcal{N}(A^H)$ , meaning the orthogonal complement of the range of  $A$  is the null space of  $A^H$ .
- $\mathbb{C}^n = \mathcal{R}(A^H) \oplus \mathcal{N}(A)$ , and  $\mathcal{R}^\perp(A^H) = \mathcal{N}(A)$ , indicating that the orthogonal complement of the range of  $A^H$  is the null space of  $A$ .

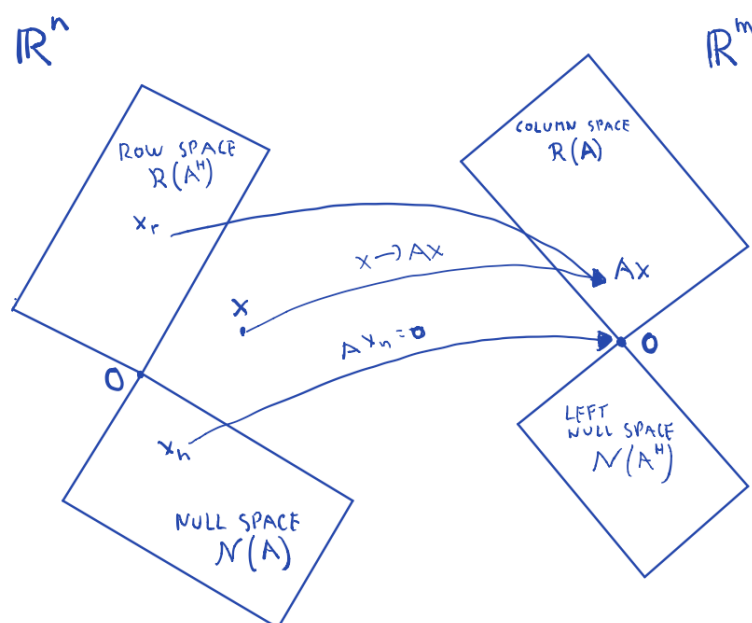


Figure 3.1: Visual representation of the four fundamental subspaces in linear algebra. The null space and row space are mapped through  $A$  and  $A^H$  to the column space and left null space respectively. Each vector in the null space maps to the zero vector in the column space, while vectors in the row space and beyond map into the column space. Would be nice to replace this with an actual diagram, will do this when I have more time

### Sample subspaces question

### Example Question 3.3.1

Consider the matrix

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

Our goal is to find the four fundamental subspaces of  $A$ : the column space  $\mathcal{R}(A)$ , the null space  $\mathcal{N}(A)$ , the row space  $\mathcal{R}(A^H)$ , and the left null space  $\mathcal{N}(A^H)$ .

#### Column Space and Null Space

Since  $A$  is already in row-echelon form, the range space (or column space) of  $A$  is spanned by the first two columns of  $A$ :

$$\mathcal{R}(A) = \text{span} \left\{ \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \right\}.$$

The dimension of the null space of  $A$  is given by the Rank-Nullity Theorem:

$$\dim(\mathcal{N}(A)) = 3 - \text{rank}(A) = 1.$$

A basis for the null space, found by solving  $A\mathbf{n} = \mathbf{0}$ , is:

$$\mathcal{N}(A) = \text{span} \left\{ \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix} \right\}.$$

### Row Space and Left Null Space

The row space of  $A^H$ , which is the column space of  $A$ , has already been determined. The left null space is the orthogonal complement of the row space in  $\mathbb{C}^m$ :

$$\mathcal{N}(A^H) = \text{span} \left\{ \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right\}.$$

Finally, the row space of  $A$  (which is the column space of  $A^H$ ) can be obtained by taking the non-zero rows of  $A$ :

$$\mathcal{R}(A^H) = \text{span} \left\{ \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \right\}.$$

## 3.4 Inverses

We classify inverses of matrices based on their dimensions and rank:

- A square and nonsingular (invertible) matrix  $A$  possesses an inverse  $A^{-1}$  such that  $AA^{-1} = A^{-1}A = I$ , where  $I$  is the identity matrix.
- Non-square matrices do not have an inverse in the traditional sense; however, they may have a left or right inverse under certain conditions:
  - A **left inverse** of a matrix  $A$  exists if there is a matrix  $B$  such that  $BA = I$  (and typically  $AB \neq I$ ), which requires  $A$  to be full column rank.
  - A **right inverse** of a matrix  $A$  exists if there is a matrix  $C$  such that  $AC = I$ , necessitating  $A$  to be full row rank.
- It is important to note that these inverses may not be unique.

## 3.5 Projections and Approximations

### Projection Theorem

### Theorem 3.5.1

Let  $V$  be a closed subspace of Hilbert space  $H$  and let  $x$  be a vector in  $H$ .

1. **Existence:** There exists  $\hat{x} \in V$  such that  $\|x - \hat{x}\| \leq \|x - v\|$  for all  $v \in V$ .
2. **Orthogonality:**  $x - \hat{x} \perp V$  is necessary and sufficient for determining  $\hat{x}$ .
3. **Uniqueness:** The vector  $\hat{x}$  is unique.
4. **Linearity:**  $\hat{x} = Px$  where  $P$  is a linear operator that depends on  $V$  and not on  $x$ .
5. **Idempotency:**  $P(Px) = Px$  for all  $x \in H$ .
6. **Self-adjointness:**  $P = P^H$ .

### Projection Matrix

### Definition 3.5.1

Given a matrix  $A$  and a vector  $\mathbf{b}$ , we seek to project  $\mathbf{b}$  onto the column space of  $A$ . The projection is denoted by  $A\mathbf{x}$ , where  $\mathbf{x}$  is the vector in the column space of  $A$  that is closest to  $\mathbf{b}$ .

### Orthogonality Principle

The difference  $\mathbf{b} - A\mathbf{x}$  is orthogonal to the column space of  $A$ , implying that it lies in the null space of  $A^T$ . Hence, we have:

$$A^T(\mathbf{b} - A\mathbf{x}) = 0 \quad (3.1)$$

## Derivation

Rearranging the above equation, we obtain:

$$A^T \mathbf{b} = A^T A \mathbf{x} \quad (3.2)$$

$$\Rightarrow \mathbf{x} = (A^T A)^{-1} A^T \mathbf{b} \quad (3.3)$$

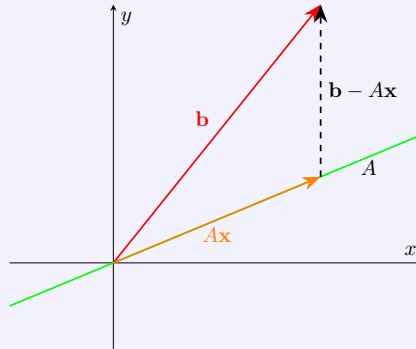
This equation provides us with the least squares solution for  $\mathbf{x}$ .

## Projection Matrix

The matrix that projects any vector  $\mathbf{b}$  onto the column space of  $A$  is known as the projection matrix,  $P$ , and is given by:

$$P = A(A^T A)^{-1} A^T \quad (3.4)$$

This projection matrix has the property that  $P\mathbf{b} = A\mathbf{x}$ , which is the projection of  $\mathbf{b}$  onto the column space of  $A$ .



## 3.6 Gram-Schmidt Orthogonalisation Process

The Gram-Schmidt process is a method for orthonormalising a set of vectors in an inner product space. It is a direct application of the projection theorem, and it helps to form an orthonormal basis, which is a cornerstone in linear algebra and has crucial applications in signal processing and systems theory.

### Orthogonal and Orthonormal Matrices

At the outset of our course, we discussed orthogonal matrices. Now, we delve into a practical use-case of these concepts.

- An **orthogonal matrix**  $A$  is a square matrix with real entries whose columns and rows are orthonormal vectors. That is, the transpose of  $A$  is also its inverse:  $A^T A = A A^T = I$ , where  $I$  is the identity matrix.
- When we construct a matrix  $A$  from a set of vectors  $\{v_i\}_{i=1}^n$  that form an orthonormal basis, it satisfies this property.
- Mathematically, if  $v_i$  for  $i = 1, 2, \dots, n$  are orthonormal, then:

$$A = \begin{pmatrix} v_{1,1} & \cdots & v_{1,n} \\ \vdots & \ddots & \vdots \\ v_{n,1} & \cdots & v_{n,n} \end{pmatrix} \quad (3.5)$$

- Such matrices not only preserve lengths (due to maintaining the induced norm) but also angles between vectors when used in linear transformations.

### The Process

The essence of the Gram-Schmidt process is to take a set of linearly independent vectors and orthogonalise them while maintaining their span.

- Start with any set of independent vectors  $\{a, b, c\}$ .
- First normalise vector  $a$  to find  $q_1$ :

$$q_1 = \frac{a}{\|a\|} \quad (3.6)$$

- To find  $q_2$ , orthogonal to  $q_1$ , project  $b$  onto  $q_1$  and subtract this projection from  $b$ , then normalise:

$$q_2 = \frac{b - \text{proj}_{q_1}(b)}{\|b - \text{proj}_{q_1}(b)\|} \quad (3.7)$$

- The projection operator  $P$  onto  $q_1$  is defined as:

$$P = \frac{aa^T}{a^T a} \quad (3.8)$$

- For vector  $c$ , ensure it is orthogonal to both  $q_1$  and  $q_2$ :

$$q_3 = \frac{c - \text{proj}_{q_1}(c) - \text{proj}_{q_2}(c)}{\|c - \text{proj}_{q_1}(c) - \text{proj}_{q_2}(c)\|} \quad (3.9)$$

- This process can be continued for higher dimensions. If there were a fourth vector  $d$ , we would subtract its projections onto  $q_1$ ,  $q_2$ , and  $q_3$  before normalizing.

## Visualising the Concept

Consider vectors  $a$ ,  $b$ , and  $c$  in a three-dimensional space. By applying the Gram-Schmidt process, we are essentially rotating and stretching the axes of our coordinate system such that each vector  $q_i$  lies along one of these new axes, ensuring orthogonality and normality.

- Visual representation of transforming vector  $b$ :

$$B = b - \frac{aa^T}{a^T a} b \quad (3.10)$$

- This transformed vector  $B$  is now orthogonal to  $a$ .
- We then apply a similar approach to vector  $c$  to ensure it is orthogonal to both  $a$  and  $B$ :

$$C = c - \frac{aa^T}{a^T a} c - \frac{bb^T}{b^T b} c \quad (3.11)$$

Make sure to normalise the vectors after their transformations!

**In general we subtract from every new vector its projections in the directions already set.** Then we make the resulting vectors orthonormal, by dividing the vectors with their magnitudes.

## Remarks

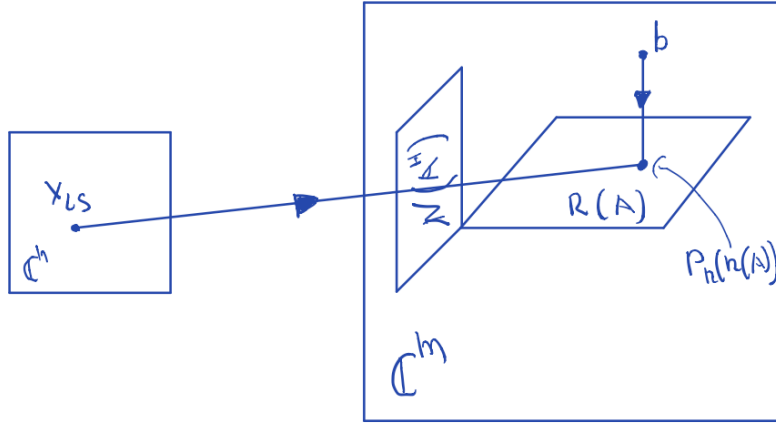
The Gram-Schmidt process is an elegant way of generating an orthogonal set from a linearly independent set. It plays a significant role in many areas of mathematics and engineering, including the understanding of Fourier series, signal processing, and the method of least squares in statistics.

## 3.7 Solutions, either exact or approximate

Let's consider the problem of finding a vector  $\mathbf{x}$  such that  $A\mathbf{x} = \mathbf{b}$ , with a particular focus on matrices  $A$  that have full column rank.

- A matrix  $A$  with full column rank is tall, implying that the columns of  $A$  are linearly independent. If a solution to the system  $A\mathbf{x} = \mathbf{b}$  exists, this solution is unique due to the full column rank of  $A$ .
- To solve the system, we aim to find the left inverse of  $A$ . This is because the left inverse allows us to compute the solution directly for a full column rank matrix.

## Geometric Interpretation to LS Solution



- The solution process involves a somewhat convoluted but geometrically inspired approach. We seek a vector  $\mathbf{x}_{LS}$  that minimizes the norm  $\|\mathbf{A}\mathbf{x}_{LS} - \mathbf{b}\|$ , making it the least squares solution. This means that  $\|\mathbf{A}\mathbf{x}_{LS} - \mathbf{b}\| \leq \|\mathbf{A}\mathbf{x} - \mathbf{b}\|$  for any vector  $\mathbf{x}$ .
- Consequently, if a solution to  $\mathbf{A}\mathbf{x} = \mathbf{b}$  exists, then  $\mathbf{x}_{LS}$  is that solution, and it can be found by projecting  $\mathbf{b}$  onto the column space of  $\mathbf{A}$ .

### Claim:

The vector  $\hat{\mathbf{x}}$  minimises  $\|\mathbf{A}\mathbf{x} - \mathbf{b}\|$  iff  $\mathbf{A}^H \mathbf{A} \hat{\mathbf{x}} = \mathbf{A}^H \mathbf{b}$ , which is the normal equation.

### Proof:

- Minimizing  $\|\mathbf{A}\mathbf{x} - \mathbf{b}\|$  is equivalent to minimizing  $\|\hat{\mathbf{b}} - \mathbf{b}\|$  where  $\hat{\mathbf{b}} \in \mathcal{R}(\mathbf{A})$ .
- By the projection theorem:  $\mathbf{b} - \hat{\mathbf{b}} \in \mathcal{R}(\mathbf{A})^\perp \Rightarrow \mathbf{b} - \hat{\mathbf{b}} \in \mathcal{N}(\mathbf{A}^H)$ .
- Therefore,  $\mathbf{A}^H(\mathbf{b} - \hat{\mathbf{b}}) = 0 \Rightarrow \mathbf{A}^H \mathbf{b} = \mathbf{A}^H \hat{\mathbf{b}} \Rightarrow \mathbf{A}^H \mathbf{A} \hat{\mathbf{x}} = \mathbf{A}^H \mathbf{b}$ .
- Conversely  $\mathbf{A}^H \mathbf{A} \hat{\mathbf{x}} = \mathbf{A}^H \mathbf{b} \Rightarrow \mathbf{A}^H(\mathbf{b} - \mathbf{A}\hat{\mathbf{x}}) = 0$ .

### 3.7.1 Example: Least-Square Deconvolution

Assume you have a system that blurs an image, this is your FIR (finite impulse response) filter. You want to recover the original sharp image. However, this observed image is noisy. Using the deconvolution process, you would attempt to reverse the blurring to approximate the original image as closely as possible.

Given the output  $\mathbf{y} \in \mathbb{C}^{n+k-1}$  of the filter and the filter is represented by matrix  $\mathbf{A} \in \mathbb{C}^{(n+k-1) \times n}$ , which is the convolution (Toeplitz) matrix associated with K-tap filter  $\mathbf{h}$ . Each row of the Toeplitz matrix is a shifted version of the filter coefficients (or taps).

If  $\mathbf{y}$  is noise-free, we can find the signal easily, but it is usually noisy. So we have the measured signal  $\mathbf{b} = \mathbf{y} + \mathbf{n}$

We can solve for  $\mathbf{A}^H \mathbf{A} \hat{\mathbf{x}} = \mathbf{A}^H \mathbf{b}$ , to get  $\hat{\mathbf{x}}$  that is the closest to the true signal  $\mathbf{x}$ .

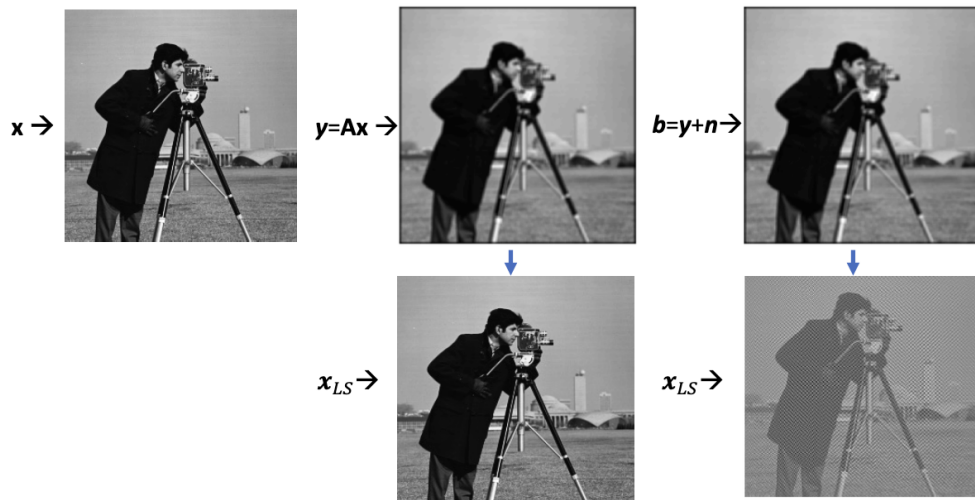


Figure 3.2: Some noise does end up back in the Least-Squares solution.

### Deconvolution:

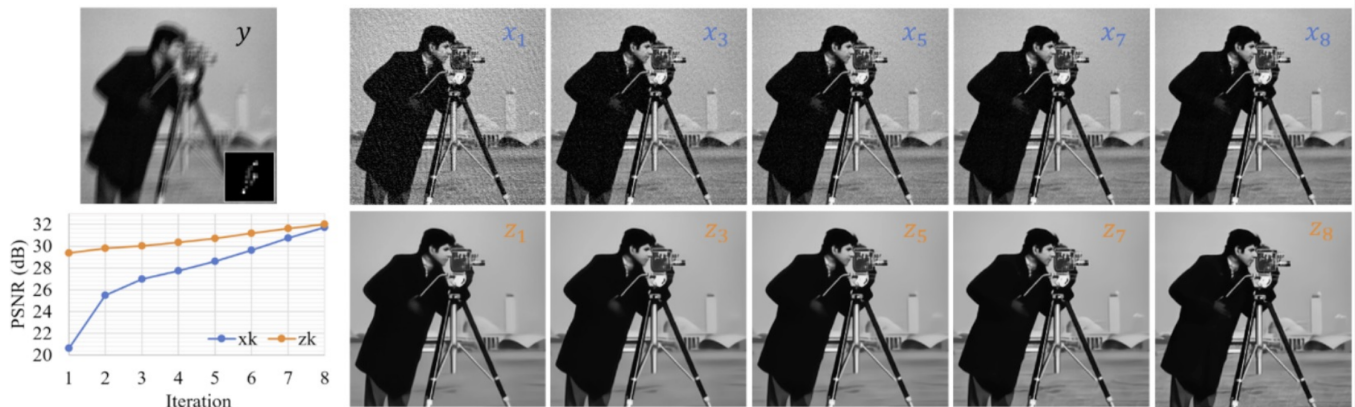


Figure 3.3: Enter Caption

## 3.7.2 Example: High-Order Regression

(Incomplete)

### 3.7.3 Solution (Full Row-Rank Case)

- We now understand how to find an approximate solution  $\mathbf{Ax} = \mathbf{b}$  in the full column rank case where a solution might exist.
- We now consider the full row rank case where  $\mathbf{A}$  is fat and we know at least one solution exists.
- When the solution exists but is not unique, the set of all solutions has a special structure.
- Let  $\mathbf{x}_p$  be a solution to  $\mathbf{Ax} = \mathbf{b}$ . Then any other solution is

$$\mathbf{x} = \mathbf{x}_p + \mathbf{z}$$

where  $\mathbf{z} \in \mathcal{N}(\mathbf{A})$ .

- Then the solution to  $\mathbf{Ax} = \mathbf{b}$  is  $S = \{\mathbf{x}_p\} + \mathcal{N}(\mathbf{A})$  where  $S$  is a translated subspace of  $\mathcal{N}(\mathbf{A})$  also known as an **affine subspace**.
- We want to pick one of the infinitely many solutions – we look for the **minimum norm solution**:

$$\min_{\|\mathbf{x}\|} \mathbf{Ax} = \mathbf{b}$$

Let  $T = \{x_p\} + S$  be an affine subspace of  $\mathbb{C}^n$ .

Then the element  $t_{MN}$  of  $T$  with minimum norm exists, is unique, and satisfies  $t_{MN} \in T \cap S^\perp$ .

Moreover,  $t_{MN} = P_{S^\perp} t$ .

First, observe that the affine subspace  $T$  can be expressed as a translation of the subspace  $S$  by the vector  $x_p$ . This translation does not affect the property of orthogonality, so the orthogonal complement of  $S$  is the same as the orthogonal complement of  $T$ .

The existence and uniqueness of  $t_{MN}$  follow from the properties of Hilbert spaces. Specifically, since  $\mathbb{C}^n$  with the standard inner product is a Hilbert space, there exists a unique element of minimal norm in the closure of the convex set  $T$ . The element  $t_{MN}$  is the projection of the origin onto the convex set  $T$ , which exists and is unique due to the Hilbert space projection theorem.

To show that  $t_{MN} \in T \cap S^\perp$ , we use the fact that for any  $t \in T$ , the difference  $t - t_{MN}$  is orthogonal to  $S^\perp$  because the projection onto a closed convex set is the nearest point in the set to the origin, and all other points in  $T$  must be further away.

Finally,  $t_{MN} = P_{S^\perp} t$  because the projection operator  $P_{S^\perp}$  is idempotent and self-adjoint, meaning that  $P_{S^\perp}$  applied to any  $t \in T$  yields the unique closest point in  $T \cap S^\perp$ , which is  $t_{MN}$ .

Therefore,  $t_{MN}$  is the minimum norm element of  $T$ , it is unique, and it lies in both  $T$  and  $S^\perp$ .

Let  $\mathbf{A}\mathbf{x} = \mathbf{b}$  and let  $A$  be a full-row rank matrix.

Amongst all the  $x \in \mathbb{C}^n$  satisfying  $\mathbf{A}\mathbf{x} = \mathbf{b}$ , there exists a unique solution with minimum norm. It lies in  $\mathcal{R}(A^H)$  and is given by  $\mathbf{x}_{MN} = \mathbf{A}_{MN} \mathbf{b}$  with  $\mathbf{A}_{MN} = \mathbf{A}^H (\mathbf{A} \mathbf{A}^H)^{-1}$ .

If  $\mathbf{x}_p$  is a solution to  $\mathbf{A}\mathbf{x} = \mathbf{b}$  then all the solutions are in the affine subspace  $T = \{\mathbf{x}_p\} + \mathcal{N}(A)$ .

Using the Dual Projection Theorem we conclude that

$$x_{MN} \in T \cap \mathcal{N}(A)^\perp = T \cap \mathcal{R}(A^H)$$

Since  $x_{MN} \in \mathcal{R}(A^H)$  implies  $x_{MN} = A^H z$  and hence  $b = Ax_{MN} = (AA^H)z$ .

Therefore,  $z = (AA^H)^{-1}b$  which implies that  $x_{MN} = A^H(AA^H)^{-1}b$ .

### 3.8 Computing Left and Right Inverses

**Claim:**

Assume  $\mathbf{A}$  is tall, full-column rank, then  $\mathbf{A}$  has a left inverse  $\mathbf{A}_{left}$

**Sketch of Proof:**

$\mathbf{A}^\top \mathbf{A}$  is full rank, so  $(\mathbf{A}^\top \mathbf{A})^{-1}$  exists. Then  $\mathbf{A}_{left} = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top$

**Likewise:**

Assume  $\mathbf{A}$  is fat and full-row rank, then  $\mathbf{A}$  has right invese  $\mathbf{A}_{right}$  and  $\mathbf{A}_{right} = \mathbf{A}^\top (\mathbf{A} \mathbf{A}^\top)^{-1}$



### 3.9 Least-Square Problems with Linear Constraints

Revisiting the minimum norm solution approach – we recall that in the full row-rank case, at least one solution exists. Since there are infinitely many solutions, we look for a specific one:

$$\min \|x\|^2 \text{ subject to } \mathbf{Ax} = \mathbf{b}$$

We can interpret the equation above differently as a least-squares minimisation with linear constraints.

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \quad b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

$$\min \|\mathbf{y} - \mathbf{Cx}\|^2 \text{ subject to } \mathbf{Ax} = \mathbf{b}$$

To solve this constrained optimisation problem, we introduce the method of Lagrangian multipliers. The Lagrangian function is given by:

$$L(\mathbf{x}, \boldsymbol{\lambda}) = \|\mathbf{y} - \mathbf{Cx}\|^2 + \sum_{i=1}^m \lambda_i (\mathbf{a}_i^\top \mathbf{x} - b_i),$$

where  $\mathbf{a}_i^\top$  are the rows of  $\mathbf{A}$  and  $\lambda_i$  are the Lagrangian multipliers associated with each constraint.

$$L(\mathbf{x}, \boldsymbol{\lambda}) = (\mathbf{y} - \mathbf{Cx})^\top (\mathbf{y} - \mathbf{Cx}) + \boldsymbol{\lambda}^\top \mathbf{Ax} - \boldsymbol{\lambda}^\top \mathbf{b}.$$

To find the minimum, we take the gradient of the Lagrangian with respect to  $\mathbf{x}$  and set it to zero:

$$\frac{\partial L(\mathbf{x}, \boldsymbol{\lambda})}{\partial \mathbf{x}} = 0 \implies 2\mathbf{C}^\top \mathbf{Cx} - 2\mathbf{C}^\top \mathbf{y} + \mathbf{A}^\top \boldsymbol{\lambda} = \mathbf{0}.$$

By combining this set of linear equations with the feasibility conditions  $\mathbf{Ax} = \mathbf{b}$ , we can write the optimality conditions as one set of  $m + n$  linear equations in the variables  $(\mathbf{x}; \boldsymbol{\lambda})$ :

$$\begin{pmatrix} 2\mathbf{C}^\top \mathbf{C} & \mathbf{A}^\top \\ \mathbf{A} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} 2\mathbf{C}^\top \mathbf{y} \\ \mathbf{b} \end{pmatrix}.$$

Solving this system yields the vector  $\mathbf{x}$  that minimises the squared norm while satisfying the constraints.

Note that the square matrix  $\begin{pmatrix} 2\mathbf{C}^\top \mathbf{C} & \mathbf{A}^\top \\ \mathbf{A} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \boldsymbol{\lambda} \end{pmatrix}$  is invertible iff  $\mathbf{A}$  is full row rank and  $\begin{pmatrix} \mathbf{C} \\ \mathbf{A} \end{pmatrix}$  is full column-rank.

## Chapter 4

# Pseudoinverses and the SVD

### 4.1 Singular Value Decomposition

Singular Value Decomposition is a fundamental technique in Linear Algebra, particularly for dealing with non-square matrices.

#### 4.1.1 Diagonalization of a Matrix

Consider a matrix  $A$  of dimension  $m \times n$  and rank  $r$ . To diagonalize  $A$ , we usually perform a transformation  $A = SAS^{-1}$ . However, this approach has limitations:

- The matrix  $A$  must be square.
- There may not be a sufficient number of eigenvectors to form the matrix  $S$ .

For example, the matrix  $\begin{pmatrix} 1 & a \\ 0 & 1 \end{pmatrix}$  with  $a \neq 0$  has only the eigenvector  $\begin{pmatrix} x \\ 0 \end{pmatrix}$ .

**Goal:** We seek a type of decomposition that applies to any matrix.

#### 4.1.2 The Singular Value Decomposition Theorem

**Theorem:**

Every matrix  $A \in \mathbb{C}^{m \times n}$  can be factored as  $A = U\Sigma V^H$ , where:

- $U$  is a unitary matrix ( $U^H U = I$ ) with columns  $u_i$  and dimension  $m \times m$ .
- $\Sigma$  is an  $m \times n$  rectangular matrix with non-negative real entries along the main diagonal.
- $V$  is a unitary matrix ( $V^H V = I$ ) with columns  $v_i$  and dimension  $n \times n$ .
- $U$  is generally different from  $V$ .
- The elements of  $\Sigma$ , called the singular values of  $A$ , are positive and typically ordered.
- For square invertible matrices, this decomposition is similar to the eigendecomposition.

#### 4.1.3 Properties of SVD

- If  $m = 2$  and  $n = 3$ ,  $\Sigma$  would look like  $\begin{pmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \end{pmatrix}$ .
- $A^H A = V \Sigma^T U^H U \Sigma V^H = V \Sigma^T \Sigma V^H$ , indicating  $\sigma_i^2$  are the eigenvalues of  $A^H A$  and  $V$  contains the eigenvectors.
- Similarly,  $AA^H = U \Sigma V^H V \Sigma^T U^H = U \Sigma \Sigma^T U^H$ , showing  $\sigma_i^2$  are the eigenvalues of  $AA^H$  and  $U$  contains the eigenvectors.
- The non-zero elements of  $\Sigma \Sigma^T$  and  $\Sigma^T \Sigma$  are identical.

#### 4.1.4 More properties of the SVD

Matrices  $A$ ,  $A^H A$  and  $AA^H$  have the same rank.

Let  $A$  be an  $m \times n$  matrix with rank  $r$ . The matrix  $A$  has  $r$  singular values. Both  $A^H A$  and  $AA^H$  have  $r$  non-zero eigenvalues which are the squares of the singular values of  $A$ . Furthermore:

- $A^H A$  is of dimension  $n \times n$ . It has  $r$  eigenvectors  $\{v_1, \dots, v_r\}$  associated with its  $r$  non-zero eigenvalues and  $n - r$  eigenvectors associated with its  $n - r$  zero eigenvalues.
- $AA^H$  is of dimension  $m \times m$ . It has  $r$  eigenvectors  $\{u_1, \dots, u_r\}$  associated with its  $r$  non-zero eigenvalues and  $m - r$  eigenvectors associated with its  $m - r$  zero eigenvalues.

We can write  $V = [v_1 \dots v_r v_{r+1} \dots v_n]$  and  $U = [u_1 \dots u_r u_{r+1} \dots u_m]$ .

Note that in the above matrices, we put first in the columns the eigenvectors of  $A^H A$  and  $AA^H$  which correspond to non-zero eigenvalues. We usually order the eigenvectors according to the magnitude of the associated eigenvalue, so the eigenvector that corresponds to the maximum eigenvalue is placed in the first column and so on.

As already shown, from  $A = U\Sigma V^H$  we obtain that  $AV = U\Sigma$  or  $A[v_1 \dots v_r v_{r+1} \dots v_n] = [u_1 \dots u_r u_{r+1} \dots u_m]\Sigma$ .

Therefore, we can break  $AV = U\Sigma$  into a set of relationships of the form  $Av_i = \sigma_i u_i$ . Note that  $\sigma_i$  is a scalar and  $v_i$  and  $u_i$  vectors.

For  $i \leq r$  the relationship  $AV = U\Sigma$  tells us that:

- The vectors  $u_1, u_2, \dots, u_r$  are in the column space of  $A$ . This observation comes directly from  $u_i = \frac{1}{\sigma_i} Av_i$ ,  $\sigma_i \neq 0$ , i.e.,  $u_i$ s are linear combinations of columns of  $A$ . Furthermore, the  $u_i$ s associated with  $\sigma_i \neq 0$  are orthonormal. Thus, they form a basis of the column space.
- The vectors  $v_1, v_2, \dots, v_r$  are in the row space of  $A$ . This is because from  $AV = U\Sigma$  we have  $U^H AV^H = U^H U \Sigma V^H \Rightarrow U^H A = \Sigma V^H \Rightarrow A^H = V \Sigma^H U^H \Rightarrow A^H = V \bar{\Sigma} U^H$ , where  $\bar{\Sigma}$  is  $\Sigma^H$  with non-zero elements  $\frac{1}{\sigma_i}$ ,  $\sigma_i \neq 0$ . Furthermore, since the  $v_i$ 's associated with  $\sigma_i \neq 0$  are orthonormal, they form a basis of the row space.
- $\mathbf{A} \mathbf{v}_i = \sigma_i \mathbf{u}_i$
- $\mathbf{v}_i^\top = \frac{1}{\sigma_i} \mathbf{u}_i^\top \mathbf{A}$
- $\mathbf{v}_i$  forms an orthonormal basis for the row space of  $\mathbf{A}$
- $\mathbf{u}_i$  forms an orthonormal basis for the column space of  $\mathbf{A}$
- Any extra  $n - r$  additional  $\mathbf{v}$ 's corresponding to zero eigenvalues of  $\mathbf{A}^H \mathbf{A}$  are taken from the null space of  $\mathbf{A}$ .
- Any extra  $m - r$  additional  $\mathbf{u}$ 's corresponding to zero eigenvalues of  $\mathbf{A} \mathbf{A}^H$  are taken from the null space of  $\mathbf{A}^H$ .

We split  $[v_1 \dots v_r v_{r+1} \dots v_n] = [V_1 \ V_2]$  and  $[u_1 \dots u_r u_{r+1} \dots u_m] = [U_1 \ U_2]$ , it then follows that:

$$A[v_1 \dots v_r v_{r+1} \dots v_n] = [u_1 \dots u_r u_{r+1} \dots u_m] \Sigma$$

- $\mathcal{R}(A) = \text{span}\{U_1\}$ ,
- $\mathcal{N}(A) = \text{span}\{V_2\}$ ,
- $\mathcal{R}(A^H) = \text{span}\{V_1\}$ ,
- $\mathcal{N}(A^H) = \text{span}\{U_2\}$ .

#### Truncated or Reduced Singular Value Decomposition

#### Definition 4.1.1

We can visualise the  $AV = U\Sigma$  as such, ignoring eigenvectors that correspond to zero eigenvalues, making  $\Sigma$  a square matrix:

$$A[v_1 \dots v_r] = [u_1 \dots u_r] \begin{bmatrix} \sigma_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_r \end{bmatrix} \Rightarrow A = u_1 \sigma_1 v_1^H + \dots + u_r \sigma_r v_r^H = A = \sum_{i=1}^r \sigma_i \mathbf{u}_i \mathbf{v}_i^H$$

- The dimension of  $A$  is  $m \times n$ .
- The dimension of  $[v_1 \dots v_r]$  is  $n \times r$ .

- The dimension of  $[u_1 \dots u_r]$  is  $m \times r$ .
- The dimension of  $\Sigma$  is  $r \times r$ .

Which is the form for the Truncated or Reduced Singular Value Decomposition, which is the sum of  $r$  rank-one matrices.

## 4.2 Relating the SVD to the pseudo-inverse

We now go back to the solution to  $A\mathbf{x} = \mathbf{b}$  when  $A$  is neither full column nor row rank.

**Claim:** The Minimum Norm Least Squares solution to  $A\mathbf{x} = \mathbf{b}$  is given by  $\mathbf{x} = A^+\mathbf{b}$  where  $A^+$  is the pseudoinverse of  $A = U\Sigma V^H$  and is given by  $A^+ = V\Sigma^+U^H$  where

$$\Sigma^+ = \begin{pmatrix} \frac{1}{\sigma_1} & & & & \\ & \ddots & & & \\ & & \frac{1}{\sigma_r} & & \\ & & & 0 & \\ & & & & \ddots \end{pmatrix}$$

**Proof:** We rewrite our problem as:

$$\min_{\mathbf{x}} \|A\mathbf{x} - \mathbf{b}\| = \min_{\mathbf{x}} \|U\Sigma V^H \mathbf{x} - \mathbf{b}\| = \min_{\mathbf{x}} \|\Sigma V^H \mathbf{x} - U^H \mathbf{b}\|$$

where the last equality follows from the fact that  $U$  is unitary.

We denote with  $\mathbf{v} = V^H \mathbf{x}$  and  $\tilde{\mathbf{b}} = U^H \mathbf{b}$  then the least-squares problem reduces to

$$\min_{\mathbf{v}} \|\Sigma \mathbf{v} - \tilde{\mathbf{b}}\|$$

for which we know the solution is  $\mathbf{v} = \Sigma^+ \tilde{\mathbf{b}}$ .

By working backwards we then find the MNLS solution

$$\begin{aligned} V^H \hat{\mathbf{x}} &= \Sigma^+ U^H \mathbf{b} \\ \Rightarrow \hat{\mathbf{x}} &= V \Sigma^+ U^H \mathbf{b} \end{aligned}$$

## 4.3 Applications of the SVD

### 4.3.1 Low Rank Approximation

Given a matrix  $D$ , the goal of a low-rank approximation is to find a matrix  $\tilde{D}$  with rank  $r$  that approximates  $D$ . This is particularly useful when  $D$  is noisy, or we know a priori that  $D$  is rank deficient.

#### Frobenius Norm

To measure the approximation, we use the Frobenius norm. For a matrix  $A$ , it is defined as:

$$\|A\|_F = \left( \sum_{i,j} |a_{i,j}|^2 \right)^{\frac{1}{2}} = (\text{trace}(A^H A))^{\frac{1}{2}} = \left( \sum_i \sigma_i^2 \right)^{\frac{1}{2}}$$

where  $\sigma_i$  are the singular values of  $A$ .

#### Singular Value Decomposition (SVD)

The Singular Value Decomposition of  $D$  is given by  $D = U\Sigma V^H$ , where  $U$  and  $V$  are unitary matrices,  $\Sigma$  is a diagonal matrix with non-negative real numbers on the diagonal (the singular values), and  $H$  denotes the conjugate transpose.

## Rank- $r$ Approximation

For a rank- $r$  approximation, we consider:

$$\tilde{D} = \sum_{i=1}^r \sigma_i \mathbf{u}_i \mathbf{v}_i^H$$

We can visualise the  $U\Sigma V^H$  as such, ignoring eigenvectors that correspond to zero eigenvalues, making  $\Sigma$  a square matrix:

$$D \begin{bmatrix} v_1 & \cdots & v_r \end{bmatrix} = \begin{bmatrix} u_1 & \cdots & u_r \end{bmatrix} \begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_r \end{bmatrix} \Rightarrow D = u_1 \sigma_1 v_1^H + \cdots + u_r \sigma_r v_r^H = \sum_{i=1}^r \sigma_i \mathbf{u}_i \mathbf{v}_i^H$$

- The dimension of  $D$  is  $m \times n$ .
- The dimension of  $\begin{bmatrix} v_1 & \cdots & v_r \end{bmatrix}$  is  $n \times r$ .
- The dimension of  $\begin{bmatrix} u_1 & \cdots & u_r \end{bmatrix}$  is  $m \times r$ .
- The dimension of  $\Sigma$  is  $r \times r$ .

This is the form for the Truncated or Reduced Singular Value Decomposition, which is the sum of  $r$  rank-one matrices.

### 4.3.2 Principal Component Analysis

Principal Component Analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components.

#### Data Matrix and Mean Centering

Consider a data matrix  $X$  with  $n$  columns and  $m$  rows, where the columns represent the number of samples and the rows correspond to  $m$  variables measured for each sample (e.g., height, weight, etc.). The preprocessing step in PCA involves mean centering, where the mean value is removed from each row.

#### Principal Components

The principal components of the data are selected so that the  $i$ -th principal component is the linear combination of the data that accounts for the  $i$ -th largest portion of the variance of the data.

#### First Principal Component

The first principal component  $a_1$  is a linear combination  $y_1^T = a_1^T X$  such that the sample variance of  $y_{1,i}$  is maximised subject to the constraint  $\|a_1\| = 1$ .

The sample variance  $\sigma_{y_1}^2$  is given by:

$$\sigma_{y_1}^2 = \frac{1}{n-1} \sum_{i=1}^n y_{1,i}^2 = \frac{1}{n-1} \sum_{i=1}^n (a_1^T x_i)^2 = a_1^T S a_1$$

where  $S$  is the covariance matrix:

$$S = \frac{1}{n-1} X X^T$$

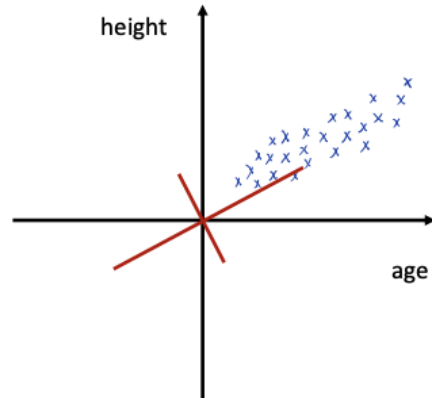
Maximising  $a_1^T S a_1$  subject to  $\|a_1\| = 1$  leads us to an eigenvalue problem, and the solution is the eigenvector corresponding to the largest eigenvalue of  $S$ . This is equivalent to the left singular vector of  $X$  related to its largest singular value.

#### Second Principal Component

The second principal component  $a_2$  is chosen so that  $y_2^T = a_2^T X$  is uncorrelated with  $y_1$ , which leads to the constraint  $a_2^T a_1 = 0$ . This vector is the singular vector related to the second largest singular value of  $X$ .

## PCA: A visual illustration

- Assume you have a data matrix  $X$  related to the age and height of 25 teenagers so  $n=25$   $m=2$ .
- The two variables are clearly correlated but you are after the principal axis
- Remove the mean from  $X$  to obtain  $X_0$
- Compute the left singular vector  $\mathbf{u}_1 = [u_{11} \ u_{21}]^T$  of  $X_0$  related to the largest singular value. Correlation is in this case positive (i.e., older kids are normally taller)
- The second principal component is also visualized



### 4.3.3 Total Least Squares

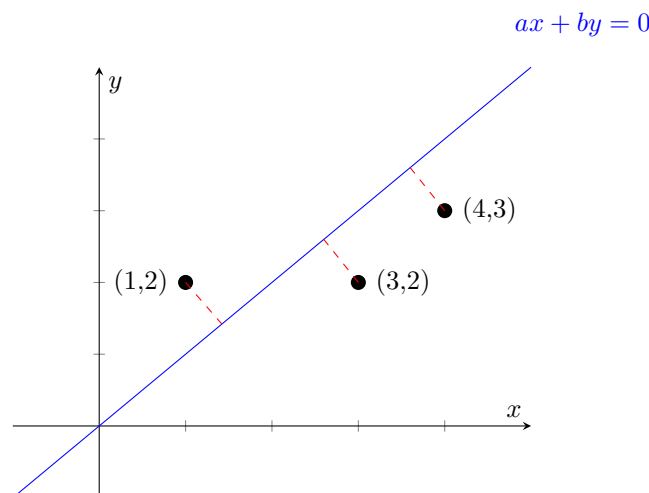
Let's look again at the minimisation of  $\|A\mathbf{x} - \mathbf{b}\|$  when  $A$  is full column rank. The LS solution essentially finds a  $\tilde{\mathbf{b}}$  such that  $\|\mathbf{b} - \tilde{\mathbf{b}}\|$  is minimised.

This is like assuming that  $A$  is noiseless and that the  $\mathbf{b}$  we observe has been corrupted by noise. The  $\mathbf{x}$  we find is the solution to  $A\mathbf{x} = \mathbf{b} + \mathbf{r}$  where  $\mathbf{r}$  is a perturbation we want to minimise.

In some cases both  $A$  and  $\mathbf{b}$  are corrupted by noise.

We then want to find a solution to the perturbed equation:  $(A + E)\mathbf{x} = \mathbf{b} + \mathbf{r}$ . It turns out that this solution is found by using the SVD.

Another way to look at the problem is to find the solution  $a, b$  to the line  $ax + by = 0$ . This line minimises the perpendicular distance between the datapoints and the line



When the solution exists is given by:

1. Compute the SVD  $[A \ \mathbf{b}] = U\Sigma V^H$
2. Then  $\hat{\mathbf{x}} = -v_{n+1}(1:n)/v_{n+1}(n+1)$

**Sketch of the proof:**

- We write  $A\mathbf{x} = \mathbf{b}$  in homogeneous form:  $C\mathbf{y} = 0$  with  $C$  being an augmented matrix of dimensions  $m \times (n+1)$ .
- $C = [A \quad \mathbf{b}]$  and  $\mathbf{y} = \begin{bmatrix} \mathbf{x} \\ -1 \end{bmatrix}$
- This evaluates to  $A\mathbf{x} - \mathbf{b} = 0$ .  $A$  is our matrix data columns, in this case it has 2 columns, one for datapoints' x-coordinates and y-coordinates.

For a given matrix  $A$  of size  $m \times n$  with  $m \geq n$ , the Singular Value Decomposition is given by  $A = U\Sigma V^H$ , where:

- $U$  is an  $m \times m$  unitary matrix,
- $\Sigma$  is an  $m \times n$  rectangular diagonal matrix with non-negative real numbers on the diagonal (the singular values),
- $V$  is an  $n \times n$  unitary matrix,
- $V^H$  is the conjugate transpose of  $V$ .

Our minimisation problem is

$$\min_{\mathbf{y}} \|C\mathbf{y}\|^2 \text{ such that } \|\mathbf{y}\| \neq 0$$

Because our formulation is  $ax + by = 0$ ,  $a$  and  $b$  are relative to each other, so we can assume that they have a magnitude of 1,  $\sqrt{a^2 + b^2} = 1$ .

Then our Lagrangian becomes:

$$\begin{aligned} \mathcal{L}(\mathbf{y}, \lambda) &= \|C\mathbf{y}\|^2 + \lambda(\|\mathbf{y}\|^2 - 1) \\ \frac{\partial \mathcal{L}}{\partial \mathbf{y}} &= 2C^H C\mathbf{y} + 2\lambda\mathbf{y} = 0 \\ &\Rightarrow C^H C\mathbf{y} = -\lambda\mathbf{y} \end{aligned}$$

From this we see that  $\mathbf{y}$  is an eigenvector of  $C^H C$ .

$$\begin{aligned} \mathcal{L}(\mathbf{y}, \lambda) &= \|C\mathbf{y}\|^2 + \lambda(\|\mathbf{y}\|^2 - 1) \\ \frac{\partial \mathcal{L}}{\partial \lambda} &= \|\mathbf{y}\|^2 - 1 = 0 \\ &\Rightarrow \mathbf{y}^H \mathbf{y} = 1 \end{aligned}$$

We combine our result from the partial derivative of  $\mathbf{y}$ , and we get this result:

$$\begin{aligned} C^H C\mathbf{y} &= -\lambda\mathbf{y} \\ \mathbf{y}^H C^H C\mathbf{y} &= -\lambda\mathbf{y}^H \mathbf{y} \\ (C\mathbf{y})^H C\mathbf{y} &= -\lambda \\ \|C\mathbf{y}\| &= -\lambda \end{aligned}$$

From this we see  $\mathbf{y}$  is a minimal eigenvalue-eigenvector pair.

- The solution is in the direction of the eigenvector corresponding to smallest eigenvalue of  $C^H C$ , this is equivalent to the column of  $V$  corresponding to smallest singular value:  $\mathbf{y} = v_{n+1}$
- The solution vector  $\mathbf{y}$  aligns with the eigenvector associated with the smallest eigenvalue of  $C^H C$ , which is the column of  $V$  corresponding to the smallest singular value. Therefore,  $\mathbf{y} = v_{n+1}$ .
- To determine the solution  $\hat{\mathbf{x}}$ , we extract the relevant components from  $v_{n+1}$ .

$$\mathbf{x} = -\frac{1}{v_{n+1}(n+1)} \begin{bmatrix} v_{n+1}(1) \\ v_{n+1}(2) \\ \vdots \\ v_{n+1}(n) \end{bmatrix} \quad (4.1)$$

where  $v_{n+1}(i)$  denotes the  $i$ -th element of vector  $v_{n+1}$ .

- In order to find  $\mathbf{x}$  we rescale  $v_{n+1}$  properly and we arrive at the solution  $\hat{\mathbf{x}} = -v_{n+1}(1:n)/v_{n+1}(n+1)$

When considering the augmented matrix  $[A \ \mathbf{b}]$ , which is formed by appending the vector  $\mathbf{b}$  as an additional column to  $A$ , the matrix now has a size of  $m \times (n+1)$ . The SVD of this augmented matrix is represented as:

$$C = [A \ \mathbf{b}] = U' \Sigma' V'^H$$

$$[A \ \mathbf{b}] \begin{bmatrix} v_1 & \dots & v_r & v_{r+1} & \dots & v_n & v_{n+1} \end{bmatrix} = \begin{bmatrix} u_1 & \dots & u_r & u_{r+1} & \dots & u_m \end{bmatrix} \Sigma$$

The dimensions and properties of the matrices in the SVD of the augmented matrix are:

- $U'$  is an  $m \times m$  unitary matrix, similar to  $U$  in the SVD of  $A$ ,
- $\Sigma'$  is an  $m \times (n+1)$  rectangular diagonal matrix, extended from  $\Sigma$  to accommodate the augmented column,
- $V'$  is an  $(n+1) \times (n+1)$  unitary matrix. The last column of  $V'$ , denoted  $v_{n+1}$ , is vital for this computation.

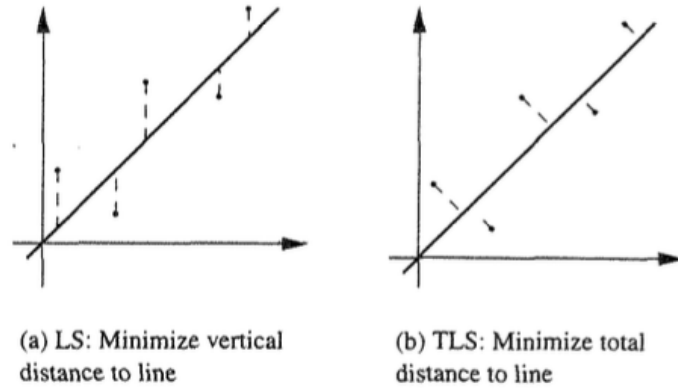


Figure 7.2: Comparison of least-squares and total least-squares fit

#### 4.3.4 Kaczmarz Algorithm

The Kaczmarz algorithm, also known as the Algebraic Reconstruction Technique (ART) in some contexts, is an iterative method for solving linear systems of equations. Its particular strength is in solving overdetermined systems, which are systems with more equations than unknowns.

The algorithm iteratively projects a solution estimate onto the solution set of each equation, one at a time, and repeats this process until convergence to a close-enough solution of linear equations  $A\mathbf{x} = \mathbf{b}$ . The procedure is as follows:

1. Initialise with any vector  $\mathbf{x}_0 \in \mathbb{R}^n$ .
2. For  $k = 1, 2, \dots$ , iter:
  - Compute the next estimate  $\mathbf{x}_{k+1}$  by projecting  $\mathbf{x}_k$  onto the hyperplane defined by the  $i$ -th equation, where  $i = k \bmod m + 1$ :

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \frac{b_i - \mathbf{a}_i^T \mathbf{x}_k}{\|\mathbf{a}_i\|^2} \mathbf{a}_i$$

3. Repeat until convergence.

Here,  $\mathbf{a}_i^T$  is the transpose of the  $i$ -th row of matrix  $A$  and represents a hyperplane in  $n$ -dimensional space. The term  $\frac{b_i - \mathbf{a}_i^T \mathbf{x}_k}{\|\mathbf{a}_i\|^2}$  computes the orthogonal projection of the residual  $\mathbf{r}_k = b_i - \mathbf{a}_i^T \mathbf{x}_k$  onto the hyperplane. This process geometrically "corrects"  $\mathbf{x}_k$  in the direction of the  $i$ -th row vector  $\mathbf{a}_i$ , moving it closer to the solution with each iteration.



### Analysis of the Approximation Error

1. Assume  $\mathbf{x}^*$  is the correct solution to  $A\mathbf{x} = \mathbf{b}$ .
2. At the  $(k+1)$ -th iteration, the approximation error is given by:

$$\mathbf{x}_{k+1} - \mathbf{x}^* = \mathbf{x}_k - \mathbf{x}^* + \frac{b_i - \mathbf{a}_i^T \mathbf{x}_k}{\|\mathbf{a}_i\|^2} \mathbf{a}_i$$

where  $\mathbf{a}_i$  is the  $i$ -th row of  $A$ , and  $b_i$  is the  $i$ -th component of  $\mathbf{b}$ .

3. To iterate, we replace  $b_i$  with  $\mathbf{a}_i^T \mathbf{x}^*$ , obtaining:

$$\mathbf{x}_{k+1} - \mathbf{x}^* = (\mathbf{x}_k - \mathbf{x}^*) - \frac{\mathbf{a}_i \mathbf{a}_i^T}{\|\mathbf{a}_i\|^2} (\mathbf{x}_k - \mathbf{x}^*)$$

4. The term  $\frac{\mathbf{a}_i \mathbf{a}_i^T}{\|\mathbf{a}_i\|^2} (\mathbf{x}_k - \mathbf{x}^*)$  represents the projection of the error  $\mathbf{x}_k - \mathbf{x}^*$  onto the direction of  $\mathbf{a}_i$ .
5. Removing the projection of the error along  $\mathbf{a}_i$  ensures that the error in the direction of  $\mathbf{a}_i$  is reduced, thus the method makes progress towards convergence.
6. At each iteration, the norm of the error does not increase:

$$\|\mathbf{x}_{k+1} - \mathbf{x}^*\| \leq \|\mathbf{x}_k - \mathbf{x}^*\|$$

This inequality shows that the Kaczmarz algorithm is convergent.

## 4.4 Summary

Scenario	Number of exact Solutions	Type of Solution	Solution
$A$ ‘tall’ and full column rank	0 or 1	Least square: $\min_{\mathbf{x}} \ \mathbf{A}\mathbf{x} - \mathbf{b}\ $	$\mathbf{x}_{ls} = (A^H A)^{-1} A^H \mathbf{b}$
$A$ ‘fat’ and full row rank	Infinitely many solutions	Minimum-norm solution: $\min_{\mathbf{x}} \ \mathbf{x}\ $ subject to $\mathbf{A}\mathbf{x} = \mathbf{b}$	$\mathbf{x}_{MN} = A^H (A A^H)^{-1} \mathbf{b}$
$A$ neither full row nor full column rank	0 or infinitely many solutions	Minimum-norm least square solution	$\hat{\mathbf{x}} = V \Sigma^+ U^H \mathbf{b}$ with $A = U \Sigma V^H$
$A$ square and full rank (trivial)	1	Unique exact solution	$\mathbf{x} = A^{-1} \mathbf{b}$

Table 4.1: Solutions to Linear Systems Based on Matrix Structure

Form	Condition	Description	Inverse	Powers	Notes
$S\Lambda S^{-1}$	Diagonalisable	A matrix is diagonalisable if it can be expressed as the product of an invertible matrix $S$ , a diagonal matrix $\Lambda$ containing the eigenvalues, and the inverse of $S$ .	$S\Lambda^{-1}S^{-1}$	$S\Lambda^k S^{-1}$	Suitable for square matrices with $n$ linearly independent eigenvectors.
$U\Sigma V^T$	Any matrix	SVD decomposes any matrix into the product of an orthogonal/unitary matrix $U$ , a diagonal matrix $\Sigma$ with singular values, and the transpose (or conjugate transpose for complex matrices) of an orthogonal/unitary matrix $V$ . $A^+$ is used for matrices that are not full rank or not invertible. It provides a 'best fit' solution to a system of linear equations.	$V\Sigma^{-1}U^T$ for invertible $\Sigma$ (all values on diagonal nonzero) $A^+ = V\Sigma^+U^H$	Generally not defined.	Applicable to all matrices, including non-square ones. When $A$ is square and full rank, $V^T = V^{-1}$ . $\Sigma^+$ is the reciprocal of values along diagonal, and zero where values are zero.
$Q\Lambda Q^T$	Symmetric Matrices	For symmetric matrices, they can be decomposed into the product of an orthogonal matrix $Q$ , a diagonal matrix $\Lambda$ , and the transpose of $Q$ .	$Q\Lambda^{-1}Q^T$	$Q\Lambda^k Q^T$	$Q$ is orthogonal, hence $Q^T = Q^{-1}$ . This is the spectral decomposition.
$PA = LU$	LU Decomposition	The matrix $A$ is decomposed with the aid of a permutation matrix $P$ , to account for pivoting, into a product of a lower triangular matrix $L$ and an upper triangular matrix $U$ .	$A^{-1} = (LU)^{-1}P = U^{-1}L^{-1}P$	Usually numerically computed	For non-integer and negative powers, typically, the LU decomposition is not used; instead, eigenvalue decomposition or other numerical methods are more suitable.

Table 4.2: Matrix Representations and Their Properties