Lecture 6

Eigenvalues and Eigenvectors

Learning Outcomes

- **Understanding Eigenvalues and Eigenvectors:**
  - Gain a comprehensive understanding of eigenvalues and eigenvectors, including their definitions, properties, and the characteristic polynomial, to comprehend the behavior of linear transformations represented by matrices.
- **Mastering Spectral Radius and Similarity Transformation:**
  - Acquire insights into the spectral radius and similarity transformation, learning their relation to the eigenvalues of a matrix and understanding how similarity transformations preserve eigenvalues, which is essential for advanced matrix analysis.
- **Delving into Normal Matrices and Diagonalization:**
  - Explore the properties and implications of normal matrices and their diagonalization, learning how normal matrices can be represented using unitary and diagonal matrices, a fundamental concept for simplifying matrix computations and analysis.

6.1 Algebraic and Geometric multiplicities

Given $A \in \mathbb{C}^{n\times n}$, λ is an eigenvalue of A if there is a $x \in \mathbb{C}^n\backslash\{0\}$ such that $Ax = x\lambda$.
Characteristic polynomial: $\rho_A(z) = \det(A - zI)$.
λ is eigenvalue if and only if $\rho_A(\lambda) = 0$.

Two quantities associated with λ,

- Algebraic multiplicity (AM) of an eigenvalue λ: largest integer q such that $(z-\lambda)^q$ is a factor of $\rho_A(z)$.
- Geometric multiplicity (GM) of an eigenvalue λ: dimension r of the kernel of $A - \lambda I$.
  Simple eigenvalue: $r = q = 1$.

Theorem 6.2. *$r \leq q$.*

Example:

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

has $\rho_A(z) = (2-z)(1-z)^3$. Eigenvalues $\lambda = 1$: Algebraic multiplicity $q = 3$, geometric multiplicity $r = 2$ since the kernel of

$$A - \lambda I_n = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

is two-dimensional.

6.3 Spectral Radius

Definition 6.1. *The spectral radius of $A \in \mathbb{C}^{n\times n}$ is*

$$\rho(A) = \max\left\{|\lambda| \,\middle|\, \lambda \text{ eigenvalue of } A\right\}.$$

Question: why do we call it the spectral radius?
Remember that $Ax = \lambda x$. Thus, there is a finite number of eigenvalues which can be denoted as a set,

$$\Lambda(A) = \lambda_0, \lambda_1, \ldots$$

This is the set of all eigenvalues corresponding to the matrix A, which is know as the spectrum of matrix A.

6.4 Similarity transformation and similar matrices

Similarity transformation: $B = M^{-1}AM$ with regular $M \in \mathbb{C}^{n\times n}$.

Theorem 6.5. *If $B \in \mathbb{C}^{n\times n}$ is similar to $A \in \mathbb{C}^{n\times n}$ then A and B have the same eigenvalues with the same multiplicities.*

Example:

$$A = \begin{pmatrix} 3 & 4 \\ 0 & 2 \end{pmatrix} \quad \text{And} \quad \tilde{A} = \begin{pmatrix} 3 & 0 \\ 0 & 2 \end{pmatrix}$$

These two matrices are also similar to,

$$\begin{pmatrix} 3 & 0 \\ 4 & 2 \end{pmatrix}, \quad \begin{pmatrix} 2 & 0 \\ 0 & 3 \end{pmatrix} \quad \begin{pmatrix} 0 & -3 \\ 2 & 5 \end{pmatrix}$$

The following similar matrices have missing eigenvector
Example:

$$\begin{pmatrix} 3 & 1 \\ 0 & 3 \end{pmatrix} \quad \text{And} \quad \begin{pmatrix} 0 & 9 \\ -1 & 6 \end{pmatrix}$$

But not similar to,

$$\begin{pmatrix} 3 & 0 \\ 0 & 3 \end{pmatrix}$$

This is because this matrix has no missing eigenvectors, while the previous two matrices have one missing eigenvector.

6.6 Normal matrices

Definition 6.1. *A matrix $A \in \mathbb{C}^{n\times n}$ is* <u>*normal*</u> *if it has n orthogonal eigenvectors.*

Normal matrices can be diagonalised: Let $Q := (q_1, \dots, q_n)$ with q_i orthonormal eigenvectors, hence Q is unitary. Then $AQ = Q\Lambda$ with $\Lambda = \operatorname{diag}(\lambda_1, \dots, \lambda_n)$ where the λ_i are the eigenvalues corresponding to the q_i. Hence

$$A = Q\Lambda Q^{-1} = Q\Lambda Q^*$$

i.e., A is similar to a diagonal matrix.
Further consequence:

$$A^*A = Q\Lambda^*Q^*Q\Lambda Q^* = Q\Lambda^*\Lambda Q^* = Q\Lambda\Lambda^*Q^* = Q\Lambda Q^*Q\Lambda^*Q^* = AA^*.$$

We will see below that normal matrices indeed can be characterised by this relation! To see this, we need

Theorem 6.7. [2.2] *Given $A \in \mathbb{C}^{n\times n}$, there is a unitary $Q \in \mathbb{C}^{n\times n}$ and an upper triangular $T \in \mathbb{C}^{n\times n}$ such that $A = QTQ^*$.*

Proof. By induction on n. In the case $n = 1$ we clearly may choose $Q = 1$ and $T = A$.
For the cases $n = 2$ where$A \in \mathbb{C}^{2\times 2}$, let y_1 denote a corresponding eigenvector with $\|y_1\|_2 = 1$ and extend it by $\{y_2\}$ to an orthonormal basis of $\mathbb{C}^2$.
Setting $U := (y_1\ y_2) \in \mathbb{C}^{2\times 2}$, the identity $Ay_1 = \lambda y_1$ leads to,

$$\begin{aligned} A\underbrace{\begin{pmatrix} y_1 & y_2 \end{pmatrix}}_{U} \\ &= \begin{pmatrix} Ay_1 & Ay_2 \end{pmatrix} \\ &= \begin{pmatrix} \lambda y_1 & Ay_2 \end{pmatrix} \end{aligned}$$

This equation can be equivalently written as,

$$\begin{aligned} AU &= U\begin{pmatrix} \lambda & \\ & U^{-1}Ay_2 \\ 0 & \end{pmatrix} \\ &= U\begin{pmatrix} \lambda & \alpha \\ 0 & \beta \end{pmatrix} \end{aligned}$$

with $U^{-1}Ay_2 = \begin{pmatrix} \alpha & \beta \end{pmatrix}^T$. Using the induction hypothesis, and noting that $\beta \in \mathbb{C}$, then

$$A = U\begin{pmatrix} \lambda & \alpha \\ 0 & \beta \end{pmatrix}U^* = \underbrace{U\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}}_{=:Q}\underbrace{\begin{pmatrix} \lambda & \alpha \\ 0 & \beta \end{pmatrix}}_{T}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}U^*$$

Following the same approach for $n \geq 2$, let y_1 denote a corresponding eigenvector with $\|y_1\|_2 = 1$ and extend it by $\{y_2, \ldots, y_n\}$ to an orthonormal basis of $\mathbb{C}^n$. Assume that the results hold for $n-1$, we claim that there is $r^* \in \mathbb{C}^{n-1}$ and $\tilde{A} \in \mathbb{C}^{(n-1)\times(n-1)}$ such that,

$$AU = U \begin{pmatrix} \lambda & r^* \\ 0 & \tilde{A} \end{pmatrix}$$

Therefore,

$$A = U \begin{pmatrix} \lambda & r^* \\ 0 & \tilde{A} \end{pmatrix} U^*$$

Using the induction assumption on $\tilde{A}$, there is a factorisation $\tilde{A} = V\tilde{T}V^*$ with upper triangular $\tilde{T} \in \mathbb{C}^{(n-1)\times(n-1)}$ and unitary $V \in \mathbb{C}^{(n-1)\times(n-1)}$. Thus,

$$\begin{aligned} A &= U \begin{pmatrix} \lambda & r^* \\ 0 & V\tilde{T}V^* \end{pmatrix} U^* \\ &= U \begin{pmatrix} 1 & 0 \\ 0 & V \end{pmatrix} \begin{pmatrix} \lambda & r^*V \\ 0 & \tilde{T} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & V^* \end{pmatrix} U^* \\ \underbrace{A\,U \begin{pmatrix} 1 & 0 \\ 0 & V \end{pmatrix}}_{Q} &= \underbrace{U \begin{pmatrix} 1 & 0 \\ 0 & V \end{pmatrix}}_{Q} \underbrace{\begin{pmatrix} \lambda & r^*V \\ 0 & \tilde{T} \end{pmatrix}}_{T} \implies \\ A &= QTQ^* \end{aligned}$$

Since U is unitary and $\begin{pmatrix} 1 & 0 \\ 0 & V \end{pmatrix}$ is also unitary, Q is unitary. Thus, this factorisation is of the required form. □

Theorem 6.8. [2.3] *If $A \in \mathbb{C}^{n\times n}$ satisfies $A^*A = AA^*$ then there is a unitary $Q \in \mathbb{C}^{n\times n}$ and a diagonal $D \in \mathbb{C}^{n\times n}$ such that $A = QDQ^*$.*

Proof. We have from Th. [2.2] that $A = QTQ^*$ with T upper triangular. We will show that T is diagonal.
Since $QT^*TQ^* = QT^*Q^*QTQ^* = A^*A = AA^* = QTQ^*QT^*Q^* = QTT^*Q^*$ we obtain $T^*T = TT^*$. Therefore

$$(T^*T)_{i,i} = \sum_{k=1}^{n} (T^*)_{i,k} T_{k,i} = \sum_{k=1}^{n} \overline{T}_{k,i} T_{k,i} = \sum_{k=1}^{i} |T_{k,i}|^2 \tag{$\star_1$}$$

where we used that T is triangular in the last identity. Similarly,

$$(TT^*)_{i,i} = \sum_{k=i}^{n} |T_{i,k}|^2. \tag{$\star_2$}$$

We now show by induction on i that the off-diagonal entries of T vanish.
For $i = 1$ we get from $(\star_1)$ and $(\star_2)$ that $|T_{1,1}|^2 = \sum_{k=1}^{n} |T_{1,k}|^2$ from which we conclude that $T_{1,k} = 0$ for $k = 2, \ldots, n$.
Let now $i > 1$ and assume that $T_{k,j} = 0$ for $1 \leq k \leq i-1$ and all $j > k$. We need to show that $T_{i,k} = 0$ for $k = i+1, \ldots, n$. Since, in particular, $T_{k,i} = 0$ for $k = 1, \ldots, i-1$ we obtain from $(\star_1)$ and $(\star_2)$ that $|T_{i,i}|^2 = \sum_{k=i}^{n} |T_{i,k}|^2$ from which we can conclude that indeed $T_{i,k} = 0$ for $k = i+1, \ldots, n$. □

Let us take as an example a 3 × 3 matrix.

$$\begin{pmatrix} t_{11} & t_{12} & t_{13} \\ 0 & t_{22} & t_{23} \\ 0 & 0 & t_{33} \end{pmatrix} \begin{pmatrix} \bar{t}_{11} & 0 & 0 \\ \bar{t}_{12} & \bar{t}_{22} & 0 \\ \bar{t}_{13} & \bar{t}_{23} & \bar{t}_{33} \end{pmatrix} = \begin{pmatrix} \bar{t}_{11} & 0 & 0 \\ \bar{t}_{12} & \bar{t}_{22} & 0 \\ \bar{t}_{13} & \bar{t}_{23} & \bar{t}_{33} \end{pmatrix} \begin{pmatrix} t_{11} & t_{12} & t_{13} \\ 0 & t_{22} & t_{23} \\ 0 & 0 & t_{33} \end{pmatrix}$$

$$\begin{pmatrix} |t_{11}|^2 + |t_{12}|^2 + |t_{13}|^2 & x & x \\ x & |t_{23}|^2 + |t_{23}|^2 & x \\ x & x & |t_{33}|^2 \end{pmatrix} = \begin{pmatrix} |t_{11}|^2 & x & x \\ x & |t_{12}|^2 + |t_{22}|^2 & x \\ x & x & |t_{13}|^2 + |t_{23}|^2 + |t_{33}|^2 \end{pmatrix}$$

Equating the two matrices we get,

$$|t_{11}|^2 + |t_{12}|^2 + |t_{13}|^2 = |t_{11}|^2 \implies |t_{12}|^2 = |t_{13}|^2 = 0$$

$$|t_{22}|^2 + |t_{23}|^2 = |t_{12}|^2 + |t_{22}|^2$$

$$|t_{33}|^2 = |t_{13}|^2 + |t_{23}|^2 + |t_{33}|^2 \implies |t_{13}|^2 = |t_{23}|^2 = 0$$

Hence T is diagonal.

Examples of Normal matrices:

- Symmetric, $A = A^T$,
- Hermitian, $A = A^*$,
- Skew symmetric, $A^T = -A$,
- Skew Hermitian, $A = -A^*$,
- Unitary matrices, $A^* = A^{-1}$

Take the following symmetric matrix as an example of a normal matrix,

$$A = \begin{pmatrix} 2 & 3 \\ 3 & 2 \end{pmatrix}$$

Then $A = Q\Lambda Q^* \iff AA^* = A^*A$ means the following,

- there exists a collection of eigenvectors that forms an orthonormal basis,
- A can be diagonalised using a unitary matrix,
- there exist an orthonormal basis of eigenvectors. In practice this means that in order to find Q, you just find your eigenbasis.

In addition exist means that you can only diagonalise with a unitary matrix.
So let us find the eigenbasis of A.

$$\begin{aligned} \rho_A(\lambda) &= \det \begin{pmatrix} 2-\lambda & 3 \\ 3 & 2-\lambda \end{pmatrix} \\ &= \lambda^2 - 4\lambda - 5 \\ &= (\lambda - 5)(\lambda + 1) \\ &\implies \lambda_{1,2} = -1, 5 \end{aligned}$$

Eigenvectors, for $\lambda_1 = -1$,

$$\rho_A(\lambda) = \begin{pmatrix} 3 & 3 \\ 3 & 3 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \implies \begin{pmatrix} x \\ y \end{pmatrix} = t_1 \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

For $\lambda_1 = 5$,

$$\rho_A(\lambda) = \begin{pmatrix} -3 & 3 \\ 3 & -3 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \implies \begin{pmatrix} x \\ y \end{pmatrix} = t_2 \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Note that $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$ is orthogonal to $\begin{pmatrix} 1 \\ -1 \end{pmatrix}$, so we just pick t_1 and t_2 to make them of length 1, thus we have an orthonormal basis of,

$$\frac{\sqrt{2}}{2} \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad \text{and} \quad \frac{\sqrt{2}}{2} \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

So our decomposition, $A = Q\Lambda Q^*$, is

$$\begin{pmatrix} 2 & 3 \\ 3 & 2 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} -1 & 0 \\ 0 & 5 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}$$

Lecture 7

Matrix Norms, Part II

Learning Outcomes

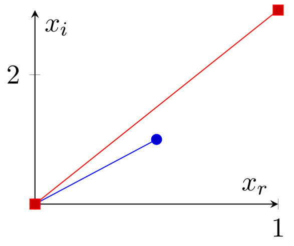
- **Advanced Comprehension of Matrix Norms:**
  - Develop a deeper understanding of matrix norms, focusing on the relationships and inequalities between spectral radius and various matrix norms, to enhance the knowledge of their properties and implications in matrix analysis.
- **Insight into Normal and Non-Square Matrices:**
  - Gain insights into the properties of normal matrices in relation to matrix norms and extend the knowledge to non-square matrices, broadening the understanding of the application of matrix norms in different types of matrices.
- **Mastery of δ-Jordan Canonical Form:**
  - Acquire knowledge of the δ-Jordan canonical form, understanding its representation of matrices and eigenvalues, its derivation, and implications, enriching the perspective on matrix representation and transformation.

Matrix Norms

Given two vectors x, and y vector norms provide information on which vector is bigger,

$$||x||_1 = |x_{11}| + |x_{21}|$$

$$||x||_2^2 = |x_{11}|^2 + |x_{21}|^2$$



Similarly for matrices, since matrices norms are based on vectors norms. Besides, matrices are linear transformation. Thus, we defined matrix norms as follows,

$$||A||_2 = \max_{||x||_2=1} ||Ax||_2$$

Also note that the set of vectors the norm maximises over is constrained to be of length 1. Another important norm, is norm of matrix power. Many real world applications require solving difference or differential equations where matrices are used as linear transformation. In addition, many machine learning algorithms are based on applying a linear transformation several times. Thus from the definition of matrix norms, the power norm can be defined as follows,

$$||A^k|| = \underset{||x||=1}{max} ||A^k x||$$

The question then is can the power matrix norm be lower bound? The following theorem states the lower bound of matrix power norms.

Theorem 7.1. [1.30] *For any matrix norm* $\|\cdot\|$, $A \in \mathbb{C}^{n\times n}$, *and* $k \in \mathbb{N}$

$$\rho(A)^k \le \rho(A^k) \le \|A^k\| \le \|A\|^k$$

Proof. Let $B := A^k$. If λ is an eigenvalue of A then λ^k is an eigenvalue of B which implies the first inequality. The third inequality is a consequence of matrix norms being sub-multiplicative. To prove the second inequality, let μ be the eigenvalue of B such that $\rho(B) = |\mu|$, let $x \in \mathbb{C}^n \setminus \{0\}$ be a corresponding eigenvector and set $X := (x, \dots, x) \in \mathbb{C}^{n\times n}$. Then,

$$\begin{aligned} Bx &= \mu x \implies \\ BX &= \mu X \end{aligned}$$

Now using the following property of matrix norm, $\|AB\| \le \|A\|\|B\|$, we get,

$$\|BX\| = |\mu|\|X\| \le \|B\|\|X\|$$

from which we get the second inequality after dividing by $\|X\|$.

$$\|B\| \ge |\mu|$$

□

The inequalities in the above theorem become equalities if the matrix is normal and we use the matrix norm induced by the Euclidean norm:

Theorem 7.2. [1.31] *If* $A \in \mathbb{C}^{n\times n}$ *is normal then* $\rho(A)^l = \|A\|_2^l$ *for all* $l \in \mathbb{N}$.

Proof. Let $x_1, \dots, x_n$ be an orthonormal basis of eigenvectors of A corresponding to the eigenvalues $\lambda_1, \dots, \lambda_n$ where w.l.o.g. $\rho(A) = |\lambda_1|$. For any $x \in \mathbb{C}^n$ we can write

$$x = \sum_{j=1}^n \alpha_j x_j \text{ with } \alpha_j = \langle x_j, x\rangle_2 \quad \Rightarrow \quad \|x\|_2^2 = \sum_{j=1}^n |\alpha_j|^2.$$

We have as well that

$$Ax = \sum_{j=1}^n \alpha_j \lambda_j x_j \quad \Rightarrow \quad \|Ax\|_2^2 = \sum_{j=1}^n |\lambda_j \alpha_j|^2.$$

Therefore

$$\frac{\|Ax\|_2^2}{\|x\|_2^2} = \frac{\sum_j |\alpha_j|^2|\lambda_j|^2}{\sum_j |\alpha_j|^2} \le \frac{\sum_j |\alpha_j|^2|\lambda_1|^2}{\sum_j |\alpha_j|^2} = |\lambda_1|^2$$

from which we see that $\|A\|_2 \le |\lambda_1|$. Together with Theorem 7.1 the assertion follows. □

Theorem 7.3. [1.32] *For all $A \in \mathbb{C}^{m\times n}$ the equality $\|A\|_2^2 = \rho(A^*A)$ holds true.*

Proof. The matrix A^*A is Hermitian, hence normal, and by Theorem 7.2

$$\rho(A^*A) = \|A^*A\|_2 = \max_{\|x\|_2=1} \|A^*Ax\|_2 = \max_{\|x\|_2=1} \big(\max_{\|y\|_2=1} \big|\langle y, A^*Ax\rangle\big|\big) \qquad (\star)$$

where we used a duality argument° for the last identity. On the one hand,

$$(\star) \ \ge \max_{\|x\|_2=1} \big|\langle x, A^*Ax\rangle\big| = \max_{\|x\|_2=1} \big|\langle Ax, Ax\rangle\big| = \max_{\|x\|_2=1} \|Ax\|_2^2 = \|A\|_2^2.$$

On the other hand, using Cauchy-Schwarz

$$\begin{aligned}(\star) \ \le \max_{\|x\|_2=1} \big(\max_{\|y\|_2=1} \big|\langle Ay, Ax\rangle\big|\big) &\le \max_{\|x\|_2=1} \big(\max_{\|y\|_2=1} \|Ay\|_2\|Ax\|_2\big) \\ &= \big(\max_{\|x\|_2=1} \|Ax\|_2\big)\big(\max_{\|y\|_2=1} \|Ay\|_2\big) = \|A\|_2^2.\end{aligned}$$

°We will not lean heavily on *duality* as part of the module, but some knowledge thereof could become useful in other contexts. As a quick introduction (you can find more in Section 1.3 of Stuart & Voss): Given a norm $\|\cdot\|$ on $\mathbb{C}^n$, the pair $(\mathbb{C}^n, \|\cdot\|)$ is a Banach space (a complete normed vector space) B. The Banach space B', the *dual* of B, is the pair $(\mathbb{C}^n, \|\cdot\|_{B'})$, where $\|x\|_{B'} = \max_{\|y\|=1} |\langle x, y\rangle|$. The usage of max here implicity relies on the fact that a continuous function on a closed, bounded set achieves its maximum value. □

Exercise: Use the above theorem to show that $\|UA\|_2 = \|A\|_2 = \|AV\|_2$ for all unitary matrices $U \in \mathbb{C}^{m\times m}$ and $V \in \mathbb{C}^{n\times n}$.

δ-Jordan canonical form

Before we start our discussion on the δ-Jordan canonical form, let us review what Jordan canonical forms are and where they actually come from. This links to our earlier discussion on similar matrices. To remind you, similar matrices are matrices that

- have the same eigenvalues.
- have the same number of eigenvectors.

We discussed that the following matrix,

$$\begin{pmatrix} 3 & 0 \\ 0 & 3 \end{pmatrix}$$

being not similar to these matrices,

$$\begin{pmatrix} 3 & 1 \\ 0 & 3 \end{pmatrix}, \quad \begin{pmatrix} 0 & 9 \\ -1 & 6 \end{pmatrix}, \quad \begin{pmatrix} 4 & -1 \\ 1 & 2 \end{pmatrix}$$

This is because the first matrix has no missing eigenvectors, while the other three matrices have one missing eigenvector. In fact, this matrix $\begin{pmatrix} 3 & 0 \\ 0 & 3 \end{pmatrix}$ is similar only to itself.

Note that the best one is $\begin{pmatrix} 3 & 1 \\ 0 & 3 \end{pmatrix}$ which is called Jordan form.

Example 1: Consider the following matrix,

$$M = \left(\begin{array}{ccc|c} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 \end{array}\right)$$

This matrix has 4 eigenvalues, $\lambda = 0, 0, 0, 0$ and two eigenvectors. Thus 2 independent eigenvectors and 2 missing.

Example 2: Consider the following matrix,

$$M = \left(\begin{array}{cc|cc} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{array}\right)$$

This matrix again has 4 eigenvalues, $\lambda = 0, 0, 0, 0$ and two eigenvectors.

However, the matrix in the first example is not similar to the one in the second. This is because the first example consists of 3×3 block and 1×1 block while the second consists of 2×2 block and 2×2 block.

These blocks are Jordan blocks. Jordan block has a repeated eigenvalue λ_i on the diagonal, 0's below the diagonal, and 1 on the top. Also, Jordan block has one eigenvector only, so the no of blocks is the no of eigenvectors.

Jordan theorem: every matrix A is similar to a Jordan matrix J where,

$$\mathrm{J} = \begin{pmatrix} \textcircled{J_1} & & & \\ & \textcircled{J_2} & & \\ & & \textcircled{J_3} & \\ & & & \textcircled{J_4} \end{pmatrix}$$

In another word, for any $A \in \mathbb{C}^{n\times n}$ there is an invertible $S \in \mathbb{C}^{n\times n}$ and a $J \in \mathbb{C}^{n\times n}$ such that $A = SJS^{-1}$ where J is a Jordan matrix, i.e., denoting by $\lambda_1, \ldots, \lambda_k$ the eigenvalues of A, J is of the form

$$J = \begin{pmatrix} J_{n_1}(\lambda_1) & 0 & \cdots & 0 \\ 0 & J_{n_2}(\lambda_2) & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & J_{n_k}(\lambda_k) \end{pmatrix}$$

with δ-Jordan blocks

$$J_{n_l}(\lambda_l) = \begin{pmatrix} \lambda_l & \delta & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \delta \\ 0 & \cdots & \cdots & 0 & \lambda_l \end{pmatrix}$$

This corresponds to the standard Jordan canonical form except that the 1s in the first off-diagonal are replaced by δs. But recall that the off-diagonal elements serve to characterise the dimensions of the eigenspaces and stand for discrete information whence we may represent this information by a number different from 1. In fact, one can derive the above δ-Jordan canonical form from the usual one by an appropriate similarity transformation.
Later on, we will pick an arbitrary small δ and use the following lemma which says that the spectral radius is no matrix norm but can be approximated by matrix norms.

Lemma 7.1. *For any $A \in \mathbb{C}^{n\times n}$ and $\delta > 0$ there is a vector norm $\|\cdot\|_\delta$ on $\mathbb{C}^n$ such that the induced norm fulfills $\rho(A) \le \|A\|_\delta \le \rho(A) + \delta$.*

Proof. The first inequality is true by Theorem 7.1.
Let $A = S_\delta J_\delta S_\delta^{-1}$ be the factorisation such that J_δ is the δ-Jordan canonical form and define the norm $\|\cdot\|_\delta$ by

$$\|x\|_\delta := \|S_\delta^{-1}x\|_\infty, \quad x \in \mathbb{C}^n$$

Then

$$\begin{aligned} \|A\|_\delta &= \max_{x\neq 0} \frac{\|Ax\|_\delta}{\|x\|_\delta} \\ &= \max_{x\neq 0} \frac{\|S_\delta^{-1}Ax\|_\infty}{\|S_\delta^{-1}x\|_\infty} \end{aligned}$$

and inserting $y = S_\delta^{-1}x$ this is

$$\begin{aligned} &= \max_{y,\, S_\delta y\neq 0} \frac{\|S_\delta^{-1}AS_\delta y\|_\infty}{\|y\|_\infty} \\ &= \max_{y\neq 0} \frac{\|J_\delta y\|_\infty}{\|y\|_\infty} \\ &= \|J_\delta\|_\infty \end{aligned}$$

and recalling that $\|\cdot\|_\infty$ is the maximum row sum we obtain that this is

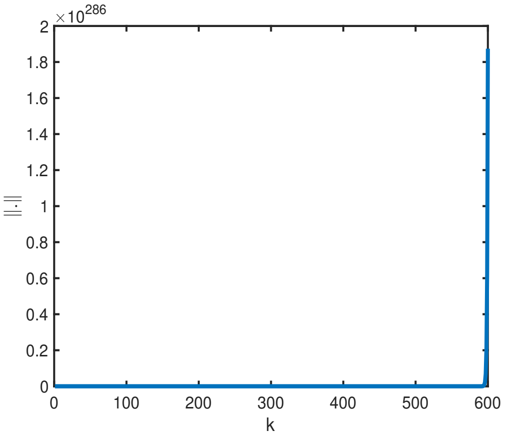
$$\begin{aligned} &= \max_i \sum_j |(J_\delta)_{i,j}| \\ &\le \max_i |\lambda_i| + \delta \\ &= \rho(A) + \delta \end{aligned}$$

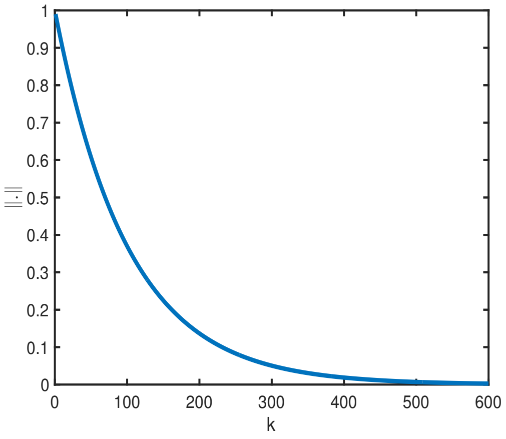
where we used the special structure of J_δ for establishing the inequality. □

Exercise 1: Consider the following two matrices,

$$A_1 = \begin{pmatrix} 0 & 3 & 0 & 0 \\ 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 \end{pmatrix}, \quad A_2 = \begin{pmatrix} \frac{30}{31} & 0 & 0 & 0 \\ 0 & \frac{60}{61} & 0 & 0 \\ 0 & 0 & \frac{80}{81} & 0 \\ 0 & 0 & 0 & \frac{100}{101} \end{pmatrix}$$

and the following two graphs,





Can you match each matrix to the correct graph of the norm of its powers.

Lecture 8

Floating point representation

Learning Outcomes

- **Mastery of Floating Point Representation:**
  - Acquire a comprehensive understanding of floating-point representation, with a focus on the binary system ($\beta = 2$) utilized by computers, learning about the approximation of numbers, the structure of mantissa, fraction bits, and exponent bits, and applying this knowledge to convert numbers into IEEE Standard 754 format.
- **Insight into Machine Precision and Numerical Error:**
  - Gain insights into machine precision (ε_m) and the relative error in numerical approximation, understanding the limitations and the range of representable values, and addressing the challenges of overflow and underflow in numerical representation.

8.1 Scientific notation and floating numbers

Floating point numbers are scientific notation. Let us consider the problem of how many meters will an object travels in 0.000005s with the speed of light, $299792458 \approx 3000000000$m/s. This requires multiplying the two numbers together. Multiplying these two numbers will incur complicated Maths. However the easy way to do it is to write the numbers in scientific notations 3.0×10^8, and 5×10^{-6}, respectively and then multiply them,

$$(3.0 \times 10^8) \times (5 \times 10^{-6}) = 15 \times 10^2$$

which is much easier, faster and understandable than trying to handle large and small numbers. In addition storing these numbers in a computer will require converting them to binary numbers. Binary representation in this case means too many 0's after the decimal and too many 1's. This results in huge number of bits to be used. However if the numbers are represented in scientific notation, storing their values will be more efficient. For example to store 0.5×10^{-5}, you will need 3 bits to represent the exponent, 1 bit to represent its sign, and to represent 0.5 only one bit is required. Finally the base will not be represented.

Any number $x \in \mathbb{R}$ can be represented with respect to a basis $\beta \in \mathbb{N}\backslash\{0, 1\}$:

$$x = \sigma \sum_{n\in\mathbb{Z}} a_n \beta^n, \qquad a_n \in \{0, \dots, \beta - 1\}\ \forall n \in \mathbb{Z}, \quad \sigma \in \{\pm 1\} \text{ sign.}$$

Computers are based on the dual system, $\beta = 2$.

Examples:

- For base 2 system, $\beta = 2$, and $a_n = \{0, 1\}$. Thus this sum becomes,

$$x = \pm(a_0 * 2^0 + a_1 * 2^1 + a_2 * 2^2 + a_3 * 2^3 + \dots)$$

- For base 10, $\beta = 10$, and $a_n = \{0, 1, \dots, 9\}$. Thus this sum becomes,

$$x = \pm(a_0 * 10^0 + a_1 * 10^1 + a_2 * 10^2 + a_3 * 10^3 + \dots)$$

However as the numbers are finite, the idea is to cut the infinite sum and to approximate x by

$$\xi = \sigma 2^e \Big(1 + \sum_{n=1}^{t} a_n 2^{-n}\Big) = \sigma 2^e \times (1.a_1 \dots a_t)_2, \qquad e = -m + \sum_{i=1}^{s} b_i 2^i.$$

The $(a_1 \dots a_t)$ are called mantissa, here of length t with fraction bits $a_n \in \{0, 1\}$, and the $(b_1 \dots b_s)$ represent the exponent of length s with exponent bits $b_i \in \{0, 1\}$. The number m is called bias or shift.

Examples,

- IEEE Standard 754 Single Precision:
  There are 32 bits to represent a number. The first bit is the sign bit, the next eight bits are the exponent bits, and the final 23 bits are the fraction bits:

$$(\sigma b_1 \dots b_8 a_1 \dots a_{23})$$

  The bias is fixed at $m = 127$.

- IEEE Standard 754 Double Precision:
  There are 64 bits to represent a number. The first bit is the sign bit, the next eleven bits are the exponent bits, and the final 52 bits are the fraction bits:

$$(\sigma b_1 \dots b_{11} a_1 \dots a_{52})$$

  The bias is fixed at $m = 1023$.

You can try this out on a particular numerical example, e.g. converting 286.75 into an IEEE Standard 754 format. In what follows we can use single (rather than double) precision just to simplify some of the algebra. Key steps:

1. Represent the decimal number in standard binary: $(286.75)_{10} = (100011110.11)_2$. This is a good opportunity for a refresher, particularly for the fractional parts which require contributions of 2^{-1} and 2^{-2} in this case in order to represent the 0.75 part of the original decimal number.

2. Normalise the binary number via binary shift (the so-called 1.m form) such that only one hidden one is left at the start: 1.0001111011.

3. Adjust with the bias for the single precision format, which for us is $2^{(8-1)} - 1 = 127$. (You can try out the double precision version as an exercise.)

4. The exponent value (+8 for 2^8 as performed in step 2) after renormalisation becomes added to the bias ($8 + 127 = 135_{10}$), which leads us to our 8−bit exponent structure being $(10000111)_2$.

5. Putting everything together (0 for the sign bit, 1000 0111 for the exponent bits (8 bits in total) and 0001 1110 1100 0000 0000 00 (23 bits in total, with padding at the end), we retrieve our final result: $(286.75)_{10} = (01000011100011110110000000000000)_{2IEEE754 \text{ single-precision}}$

Additional resource: if you would like to try out some examples (or verify your own calculations) there are nice converters out there, as well as other worked-out cases of various degrees of complexity.

8.2 Errors in floating point representation

In base 10 we have,

$$(\dots \quad 100 \quad 10 \quad 1 \quad . \quad \tfrac{1}{10} \quad \tfrac{1}{100} \quad \tfrac{1}{1000} \quad \dots)$$

Thus $\frac{1}{10}$ in decimal is equal to 0.1 . In base 2 system,

$$(\dots \quad 4 \quad 2 \quad 1 \quad . \quad \tfrac{1}{2} \quad \tfrac{1}{4} \quad \tfrac{1}{8} \quad \dots)$$

thus, 0.1 is represented as, $0.0\overline{001}$ which should continue to ∞, i.e

$$(0.1)_2 = 0.000110011001100110011\dots$$

But remember, floating point representation is basically scientific notation in bases 2. In this representation, since the number is finite, the number of bits is truncated to a finite number (32-bits computers only store 23 significant digits for example). And this is the problem, because in this case precision will be lost. This means that computers do not understand recursions.

Example: what is $\frac{1}{10} + \frac{3}{10}$?

$$\frac{1}{10} = 0.00011001100110011001100$$
$$\frac{2}{10} = 0.01001100110011001100110$$

So a computer will say that $\frac{1}{10} + \frac{2}{10} \neq \frac{3}{10}$

The relative error when approximating a number x by its nearest neighbour ξ is

$$\frac{|\xi - x|}{|x|} \leq \varepsilon_m \approx 10^{-16}.$$

ε_m is called machine precision. For positive x, values for ξ are between $\approx 10^{-320}$ and 10^{308}. If x is bigger (smaller) then we have to deal with an overflow (underflow).

8.3 Landau (Big O) Notation

Landau (Big O) Notation. Asymptotic notation is a useful tool in assessing algorithmic performance, as well as analysing (and indeed constraining) error levels in implementation. You may have come across descriptions of algorithms (in terms of operation count) as being linear/polynomial/exponential in the relevant variable n (e.g. the size of a matrix) or having a concrete estimate as being $\mathcal{O}(n)$, $\mathcal{O}(n^2)$, $\mathcal{O}(n \log n)$ etc.

In general form, if we let f be a real- or complex-valued function and g be a real-valued function on some unbounded subset of the positive real numbers, with $g(x)$ strictly positive for sufficiently large values of x. Then we write $f(x) = \mathcal{O}(g(x))$ as $x \to \infty$ if there exists a positive $M \in \mathbb{R}$ and a $x_0 \in \mathbb{R}$ such that $|f(x)| \leq Mg(x) \ \forall \ x \geq x_0$.

In more practical terms, the $\mathcal{O}(g(n))$ provides a worst-case scenario estimate for the runtime of an algorithm (in that it is bounded between 0 and $Mg(x)$. As a concrete example, we can say that if $f(x) = 5x^3 + 6x + 2$ then $f = \mathcal{O}(x^3)$ given the cubic term provides the dominant route towards increase as $x \to \infty$. Similarly, there exists a lower bound equivalent (denoted by Ω).

Finally, this framework also provides the means to create an asymptotic tight bound using the Θ-notation, which sets a proportionality relationship between the two relevant functions. In other words, $f(x) = \Theta(g(x))$ means that there exist positive constants c_1, c_2 and x_0 such that $0 \leq c_1 g(x) \leq f(x) \leq c_2 g(x) \ \forall \ x \geq x_0$.

Additional resources and examples: freeCodeCamp offers some nice background material and visualisations on the topic if you are keen to develop your intuition on the topic.

Exercise: Convert the following into an IEEE Standard 754 format number into decimal

11000011010101100000000000000000

Exercise: Formulate an algorithm that computes the factorial of a number $x \in \mathbb{R}$, then evaluate the big O that upper bounds this function.