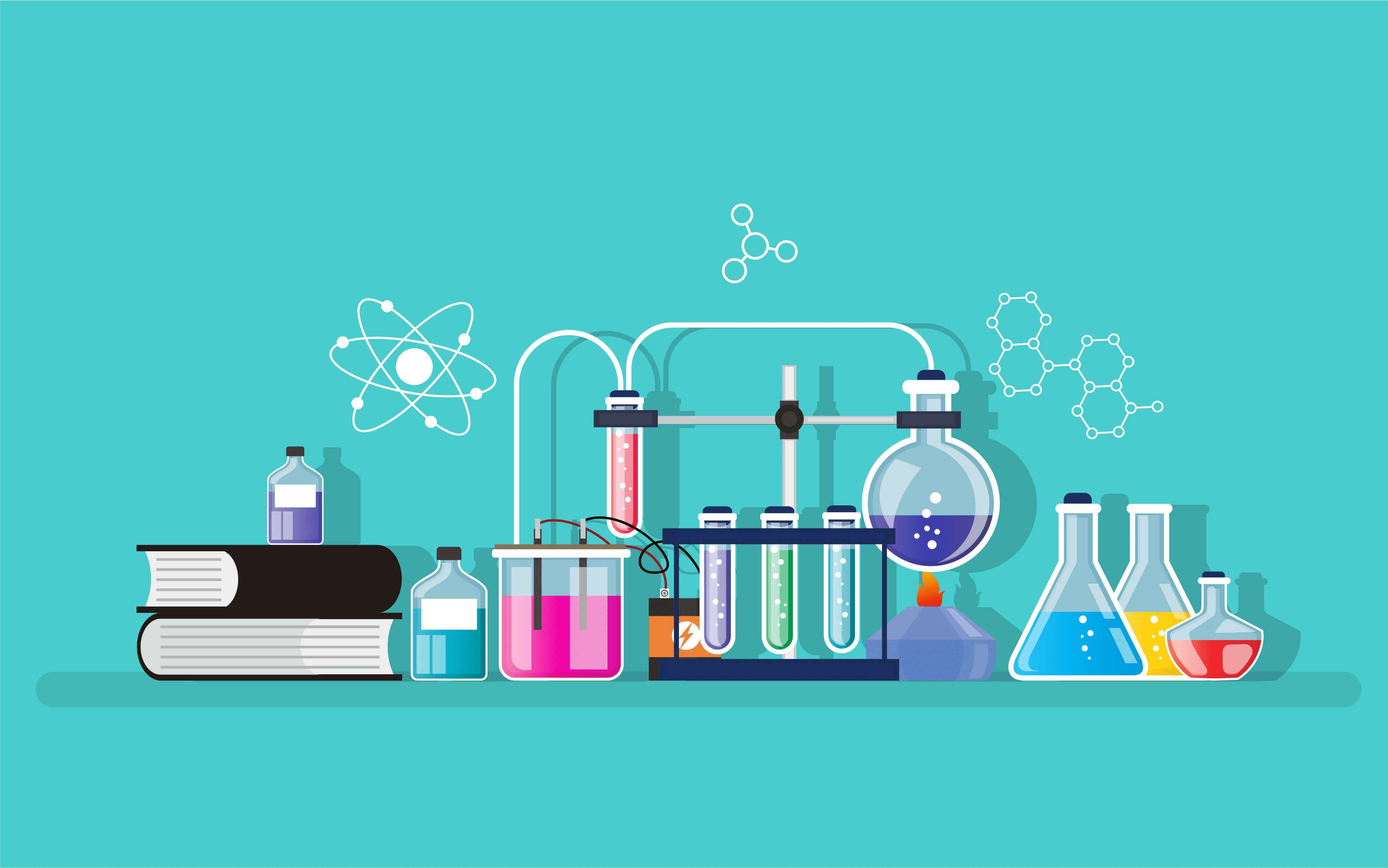
# **Chapter Two**

# **Chemistry in Vector Space**



# **2.1 What is Chemistry?**

Chemistry is the scientific study of matter, including its qualities, composition, and structure, as well as the changes that occur during chemical reactions. It is known as the core science because it links and overlaps with many other scientific fields, including as physics, biology, and environmental science. Chemists study the behavior of atoms and molecules and use that knowledge to manipulate and control matter to produce new substances or enhance current ones.

# **2.2 Application of Chemistry in Vector Space**

Vector spaces have several applications in chemistry, particularly in the fields of quantum chemistry, spectroscopy, and chemical informatics. Here are some key applications:

**Quantum Chemistry:** Quantum chemistry deals with the electronic structure of molecules and atoms. In this field, vector spaces are extensively used to represent the wave functions of electrons and molecular orbitals. The basis functions for these vector spaces are typically atomic orbitals or other mathematical functions. Quantum chemistry calculations involve operations on these vectors to determine properties such as energy levels, bond lengths, and electron density distributions.

**Spectroscopy:** In spectroscopy, vector spaces are employed to analyze and interpret the spectra of chemical compounds. Different types of spectroscopy, such as nuclear magnetic resonance (NMR), infrared (IR), and ultraviolet-visible (UV-Vis) spectroscopy, generate complex data that can be represented and analyzed in vector spaces. Techniques like Fourier transform and principal component analysis (PCA) use vector space mathematics to simplify and extract meaningful information from spectral data.

**Chemoinformatics:** Chemoinformatics involves the storage, retrieval, and analysis of chemical data. Chemical compounds can be represented as molecular graphs or vectors, where each element or substructure is assigned a unique identifier or feature vector. These representations allow for efficient searching of chemical databases, similarity analysis, and structure-activity relationship (SAR) studies.

**Crystallography:** Vector spaces are used in crystallography to describe the three-dimensional arrangement of atoms in a crystal lattice. X-ray crystallography data is typically represented as electron density maps in a three-dimensional grid, which can be treated as a vector space.

**Chemical Modeling:** Computational chemistry often involves the creation of mathematical models to predict chemical behavior. These models may use vector spaces to represent chemical reactions, molecular properties, or chemical kinetics. Linear algebraic techniques can help solve systems of equations, optimize molecular geometries, and simulate chemical processes.

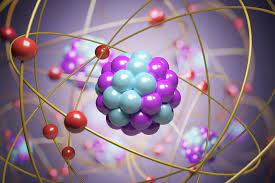
**Chemical Data Analysis:** Analyzing large datasets of chemical information often involves dimensionality reduction and data visualization techniques like principal component analysis (PCA) and multidimensional scaling (MDS), which are based on vector space representations. These methods help researchers identify patterns and relationships within complex chemical data.

**Molecular Dynamics Simulations:** In molecular dynamics simulations, molecules are modeled as collections of atoms interacting through forces. The positions and velocities of atoms are often represented as vectors, and numerical integration techniques are used to simulate the time evolution of these vectors to study molecular behavior over time.

**Chemical Engineering:** Vector spaces can be applied in chemical engineering for process optimization, control, and modeling. They can be used to represent various parameters in chemical processes, such as temperature profiles, concentration gradients, and flow rates, and to perform optimization and sensitivity analysis.

In summary, vector spaces play a crucial role in various aspects of chemistry, from quantum chemistry to data analysis and modeling. They provide a powerful mathematical framework for representing and analyzing complex chemical systems and data.

**2.3 Quantum Chemistry**



## **2.3.1 What is Quantum Chemistry**

Quantum chemistry deals with the behavior of electrons and nuclei within molecules and atoms, and it relies heavily on the principles of quantum mechanics.

The birth of quantum chemistry is often recognized as starting with the discovery of the Schrödinger equation and its application to the hydrogen atom. However, a 1927 article by Walter Heitler and Fritz London is often recognized as the first milestone in the history of quantum chemistry. This was the first application of quantum mechanics to the diatomic hydrogen molecule, and thus to the phenomenon of the chemical bond. Here are some key aspects of quantum chemistry

**Schrödinger Equation:** This equation, which explains how a quantum system's wave function varies over time, is fundamental to quantum chemistry.   
The time-independent Schrödinger equation is written as:

(Psi) is the wave function, which describes the quantum state of the system.

H is the Hamiltonian operator, which represents the total energy of the system.

E represents the energy of the system.

**Quantum Mechanics:** Quantum mechanics is a fundamental theory in physics that describes the behavior of matter and energy at the smallest scales, such as atoms and subatomic particles. It is based on the principles of wave-particle duality and the Schrödinger equation, which provides a mathematical framework for understanding the behavior of quantum systems.

**Electron Distribution:** Quantum chemistry is primarily concerned with understanding the distribution of electrons in atoms and molecules. It calculates electron densities, electron clouds, and molecular orbitals to describe the probability of finding electrons in different regions of space.

Some examples of quantum chemistry:

Hydrogen Atom: One of the simplest examples is the hydrogen atom, where quantum chemistry accurately predicts the energy levels and electron distribution in the hydrogen atom.

Water Molecule: Quantum chemistry can explain the structure of the water molecule, the nature of its chemical bonds, and its various properties.

Benzene: Quantum chemistry can elucidate the electronic structure of benzene, its stability, and its unique ring of conjugated π-electrons.

Enzyme-Substrate Interactions: Quantum chemistry is used to study the interactions between enzymes and substrates in biochemistry, providing insights into enzymatic mechanisms and catalysis.

# **2.3.2 Relation between Quantum Chemistry and Vector Space**

**Hilbert Space:** In quantum mechanics, the state of a physical system, such as an electron in an atom, is described by a complex-valued wave function, often denoted as ψ (psi). These wave functions exist in a mathematical space called a Hilbert space. A Hilbert space is a complex vector space with an inner product (a way to calculate the angle between vectors) that satisfies certain properties. The wave function itself is essentially a vector in this Hilbert space.

**Wave functions:** Quantum states of electrons and nuclei in molecules are described by wave functions, which are complex-valued functions of the spatial coordinates. These wave functions belong to a vector space, often referred to as the Hilbert space. Molecular orbitals and the electron distributions within molecules are represented by linear combinations of wave functions.

In quantum chemistry, vector spaces are fundamental for representing the wave functions of electrons and molecular orbitals. These wave functions are typically represented as vectors in a mathematical vector space. Let's explore this concept with a simple example.

**Example**: Representing Electron Spin States

In quantum chemistry, one of the most fundamental vector spaces is the space of electron spin states. Electrons have two possible spin states: "up" and "down," often represented as |↑⟩ and |↓⟩. These two states form a basis for a two-dimensional vector space, which we can represent as a complex vector space.

In this vector space, a general electron spin state can be represented as a linear combination of the basis states:

|ψ⟩ = α|↑⟩ + β|↓⟩

Here, α and β are complex numbers (coefficients), and |ψ⟩ represents an arbitrary electron spin state. This is an example of a superposition state where an electron can exist in a linear combination of both spin states.

Let's say we have an electron in a spin state |ψ⟩, and we want to find the probability of measuring it in the "up" state |↑⟩. To do this, we calculate the absolute square of the coefficient α:

P(↑) = |α|^2

The probability of measuring the electron in the "down" state |↓⟩ would be:

P(↓) = |β|^2

The coefficients α and β must satisfy the normalization condition:

|α|^2 + |β|^2 = 1

This ensures that the probabilities of finding the electron in either spin state add up to 100%.  
Here's a simple numerical example:

Suppose we have an electron in the spin state:

|ψ⟩ = (1/√2)|↑⟩ + (i/√2)|↓⟩

In this case, α = 1/√2 and β = i/√2. We can calculate the probabilities:

P(↑) = |α|^2 = (1/√2)^2 = 1/2

P(↓) = |β|^2 = (i/√2)^2 = -1/2

Notice that the probabilities add up to 1, satisfying the normalization condition. This means there is a 50% chance of measuring the electron in the "up" state and a 50% chance of measuring it in the "down" state.

This example demonstrates the use of a vector space to represent electron spin states and calculate probabilities, which is a fundamental concept in quantum chemistry.

**Operators and Matrices:** Linear operators are used to express quantum observables such as location (x), momentum (p), and energy (H, the Hamiltonian). Matrix representations are commonly used to depict these operators in the context of quantum chemistry. Linear algebra is used to manipulate and calculate the effects of these operators on the quantum state, leading to observable results.

**Position Operator (x):** The position operator, denoted as "x," is an operator that describes the position of a particle. In one dimension, the position operator acts on a wave function ψ(x) as follows:

xψ(x) = xψ(x)

This equation represents the action of the position operator on the wave function. In matrix form, if you discretize the position space into small intervals, x becomes a diagonal matrix:

[xψ(x)] = [x₁ψ(x₁) x₂ψ(x₂) ... xₙψ(xₙ)]

The eigenvalues of this matrix correspond to the possible positions of the particle, and the eigenvectors represent the wave function at those positions.

**Momentum Operator (p):** The momentum operator, denoted as "p," is another key operator in quantum mechanics. In one dimension, the momentum operator acts on a wave function ψ(x) as follows:

pψ(x) = -iħ(∂ψ(x)/∂x)

In matrix form, this operator becomes a derivative matrix that operates on the discretized wave function:

[pψ(x)] = [-iħ(∂ψ(x₁)/∂x) -iħ(∂ψ(x₂)/∂x) ... -iħ(∂ψ(xₙ)/∂x)]

Eigenvalues of this matrix represent the possible momenta of the particle.

**Hamiltonian Operator (H):** The Hamiltonian operator represents the energy of a quantum system. It is often expressed as a sum of kinetic and potential energy terms and acts on the quantum state to give the total energy.

Hψ(x) = (-ħ²/2m)∂²ψ(x)/∂x² + V(x)ψ(x)

The Hamiltonian operator is represented as a matrix in a discretized space, and its eigenvalues correspond to the possible energy levels of the system.

**Matrices in Quantum Chemistry:** Matrices are used to represent operators and transformations in quantum chemistry. The matrix representation of operators allows for the manipulation and calculation of observables. Here are some examples:

Matrix Representation of Operators: To represent an operator A in matrix form, we often use a basis set of functions (e.g., atomic orbitals) and calculate the matrix elements. The matrix elements of A are given by:

Aᵢⱼ = ⟨ψᵢ|A|ψⱼ⟩

For instance, the matrix representation of the position operator x in a basis set could be derived by calculating the expectation values for x between different basis functions.

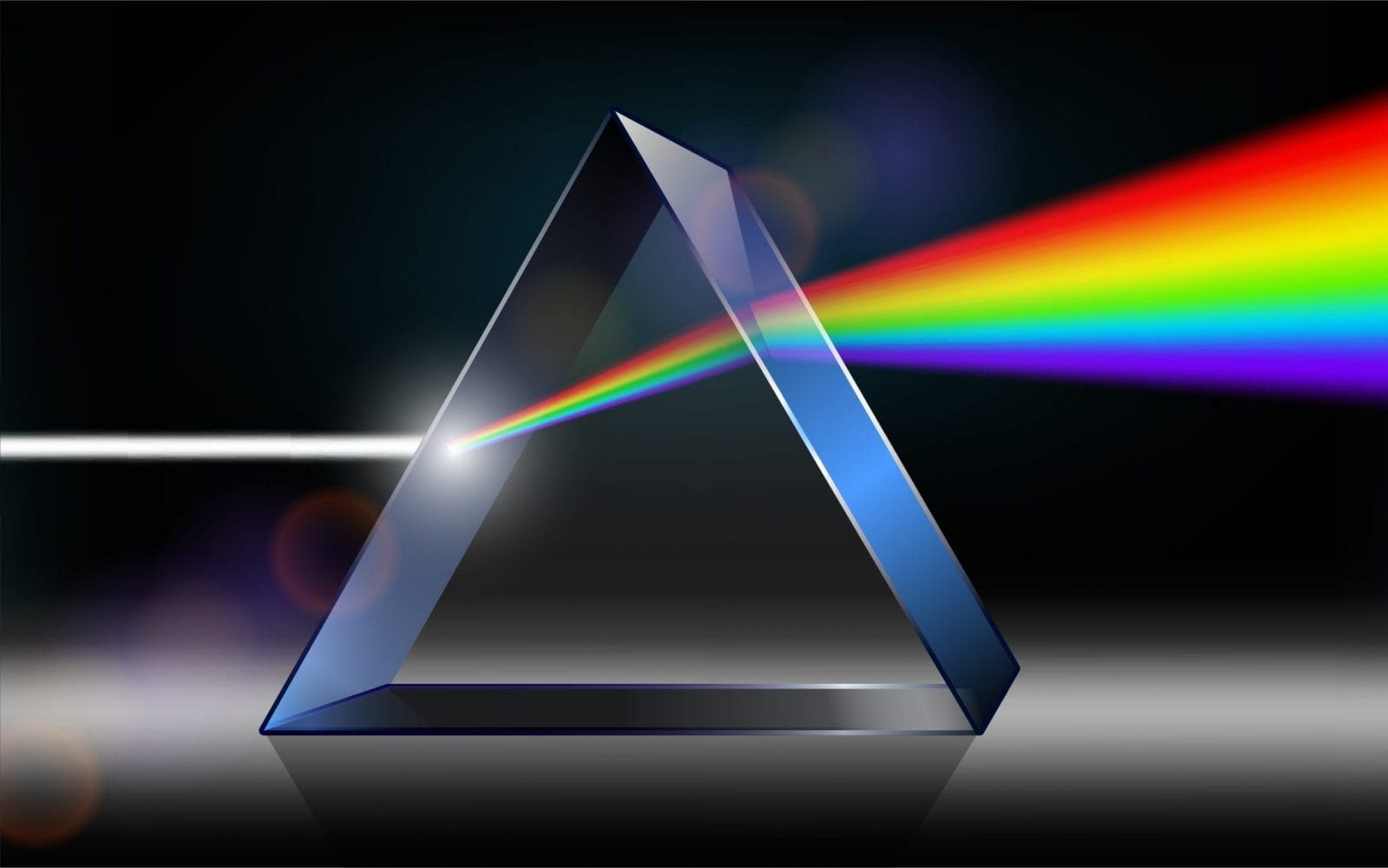
**Matrix Diagonalization:** Diagonalization of a matrix is often used to find the eigenstates and eigenvalues of operators. For instance, diagonalizing the Hamiltonian matrix yields the energy eigenvalues and the corresponding wave functions for a quantum system.

**Eigenstates and Eigenvalues:** Eigenstates and Eigenvalues are fundamental concepts related to the mathematical representation of quantum systems using vector spaces. The behavior of quantum systems, like atoms and molecules, must be understood in light of these ideas.

Eigenstates: Eigenstates, also known as eigenvectors, are vectors in a vector space that represent the quantum states of a physical system. In quantum chemistry, the vector space often corresponds to the Hilbert space, which is a complex, infinite-dimensional space where quantum states reside.  
An eigenstate of an operator represents a specific quantum state that remains unchanged when operated upon by that operator. Mathematically, if |ψ⟩ is an eigenstate of an operator A, then applying operator A to |ψ⟩ results in a scalar multiple of |ψ⟩: A|ψ⟩ = λ|ψ⟩, where λ is the eigenvalue associated with the eigenstate |ψ⟩.  
Eigenstates are used to describe the possible quantum states of a system, such as the energy levels of electrons in an atom or the molecular orbitals in a molecule.

In conclusion, vector spaces are used by quantum mechanics to give a mathematical framework for comprehending the behavior of quantum systems. It enables the description of quantum states, the calculation of probabilities, and the representation of quantum operators, all of which are essential for practical applications in fields such as quantum computing, quantum communication, and quantum chemistry.

# **2.4 Spectroscopy**



## **2.4.1 What is Spectroscopy**

Spectroscopy is a branch of science that deals with the study of the interaction between matter and electromagnetic radiation. In the context of chemistry, spectroscopy is a powerful analytical technique used to investigate the composition, structure, and properties of matter. It is particularly valuable for identifying and quantifying the chemical components in a sample, as well as for probing various aspects of molecular and atomic behavior.

The basic principle of spectroscopy involves the interaction of matter with light (or other forms of electromagnetic radiation) across a range of wavelengths. When light interacts with a sample, it can be absorbed, transmitted, or scattered, depending on the nature of the sample and the wavelength of the light. By measuring the changes in the intensity or frequency of the light as it interacts with the sample, scientists can obtain information about the sample's properties.

## **2.4.2 Relation between Spectroscopy and Vector Space**

In spectroscopy, a scientific method for examining how matter interacts with electromagnetic radiation, vector spaces are essential. Spectroscopy is employed in a wide range of fields, including chemistry, physics, astronomy, and environmental science. Here are some key applications of vector spaces in spectroscopy:

**Data Representation:** Spectroscopy usually involves taking measurements at several wavelengths or frequency points and recording the intensity or response at each one. This data can be represented as a set of numerical values, often organized as a vector, where each component of the vector corresponds to the intensity or response at a specific wavelength or frequency. This makes it possible to visualize the data from a spectrum as a point in a vector space with high dimensions.

**Spectral Analysis:** Spectra are often represented as vectors where each element corresponds to the intensity of light at a specific wavelength or frequency. Mathematical methods such as linear algebra can be used to manipulate and analyze these spectral vectors. Principal Component Analysis (PCA) and Singular Value Decomposition (SVD) are examples of techniques that utilize vector spaces to reduce data dimensionality and extract meaningful information from complex spectra.

**Quantum Mechanics and Molecular Spectroscopy:** In the field of quantum mechanics, the wave functions that describe the energy states of atoms and molecules are represented as vectors in a complex vector space. Spectroscopy is used to determine the energy levels and transitions between these levels, which are represented mathematically as vectors. The vector space formalism allows for the accurate prediction and analysis of spectral lines and transitions.

**Chemical Analysis:** In analytical chemistry, vector spaces are used to represent the spectra of chemical compounds. Spectroscopic techniques like infrared spectroscopy, nuclear magnetic resonance (NMR) spectroscopy, and mass spectrometry generate spectra that are treated as vectors. These spectra are then compared to reference spectra in a database to identify and quantify compounds.

**Image Spectroscopy:** Hyperspectral imaging, a technique used in remote sensing and Earth observation, involves capturing images at hundreds of narrow, contiguous spectral bands. Each pixel in such images can be considered a vector in a high-dimensional space. Vector space techniques are applied to process and analyze these data, identifying materials, detecting pollutants, and monitoring environmental changes.

**Astronomical Spectroscopy:** In astronomy, the light from celestial objects is dispersed into spectra to reveal information about their composition, temperature, and motion. Spectral data from telescopes and spectrometers are often represented as vectors. Vector space methods are applied to classify stars, identify elements in stellar atmospheres, and analyze the redshift of galaxies, which indicates their relative motion.

**Signal Processing:** In various spectroscopic techniques, signal processing methods are used to enhance the quality of the data. Techniques like Fourier Transform and cross-correlation involve manipulating data as vectors to extract useful information.

In summary, vector spaces are fundamental in spectroscopy for the representation, analysis, and interpretation of spectral data, as well as for solving complex problems in various scientific and industrial applications.

# **2.5 Chemoinformatics**

****

## **2.5.1 What is Chemoinformatics**

Chemoinformatics is a major discipline in theoretical chemistry that uses artificial intelligence and data sciences to tackle current social and innovation challenges in Chemistry. It concerns the development, creation, organization, storage, dissemination, analysis, visualization and use of chemical information. Chemoinformatics provides computer methods for learning from chemical data and for modeling tasks a chemist is facing. The field has evolved in the past 50 years and has substantially shaped how chemical research is performed by providing access to chemical information on a scale unattainable by traditional methods.

Chemoinformatics is used to predict the toxicity of chemical substances, which is critical for safety evaluation in medication development and environmental investigations. Environmental chemoinformatics is concerned with forecasting the environmental destiny and consequences of chemicals, such as their persistence, bioaccumulation, and toxicity in ecosystems.

## **2.5.2 Relation Between Chemoinformatics and Vector Space**

Chemoinformatics and vector space are related because vector space is used to represent chemical structures in chemoinformatics. The similarity between two chemical structures can be measured by the distance between their corresponding vectors in this space. This concept has broad conceptual and practical applicability in many areas of chemistry, including drug design and discovery.

Chemical Structure Representation: Molecules are represented as mathematical vectors in a high-dimensional space in chemoinformatics. Each atom or molecular characteristic is assigned a numerical number or identification, which creates the vector's components or dimensions. A molecule, for example, can be represented as a binary vector, where each dimension corresponds to one atom in the molecule and the value of each dimension signals whether that atom is present (1) or absent (0).

Molecular Descriptors: Molecular descriptors are quantitative representations of various molecular properties, such as size, shape, electronegativity, and functional groups. These descriptors are often computed and represented as vectors. Each descriptor corresponds to a dimension in the vector space, and its value quantifies a specific aspect of the molecule. For instance, the molecular weight of a compound could be a dimension in the vector, and its value is the molecular weight of the molecule.

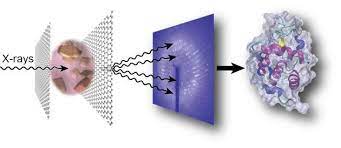
Similarity and Distance Metrics: In chemoinformatics, one common task is to measure the similarity or dissimilarity between chemical compounds. This is done by comparing their vector representations. Various distance metrics are employed, such as the Euclidean distance or cosine similarity, to quantify the difference between vectors. Similar compounds will have vectors that are closer to each other in this vector space.

Dimensionality Reduction: Vector space representations of chemical compounds can be high-dimensional due to the large number of descriptors or features used. High dimensionality can make analysis and visualization challenging. Dimensionality reduction techniques, such as Principal Component Analysis (PCA) or t-Distributed Stochastic Neighbor Embedding (t-SNE), are applied to project the vectors into lower-dimensional spaces while preserving the most relevant information. This makes it easier to visualize and analyze the data.

Chemoinformatics Databases: Chemoinformatics databases, which store information about chemical compounds, often utilize vector representations for efficient data storage and retrieval. Each compound's properties, structure, and other information are stored as vectors. These representations enable rapid searching and retrieval of compounds based on specific criteria or similarity to a reference compound, all within the vector space.

In summary, the relationship between chemoinformatics and vector space is evident in the use of mathematical vectors to represent and analyze chemical compounds and properties.

**2.6 Crystallography**



## **2.6.1 What is Crystallography?**

Crystallography is a branch of science that deals with the study of the arrangement of atoms in crystalline solids and the geometric structure of crystal lattices. It is a fundamental subject in the fields of materials science and solid-state physics. The word crystallography is derived from the Ancient Greek word κρύσταλλος (krústallos; “clear ice, rock-crystal”), with its meaning extending to all solids with some degree of transparency, and γράφειν (gráphein; “to write”).

Before the development of X-ray diffraction crystallography, the study of crystals was based on physical measurements of their geometry using a goniometer. This involved measuring the angles of crystal faces relative to each other and to theoretical reference axes (crystallographic axes), and establishing the symmetry of the crystal in question. The position in the 3D space of each crystal face is plotted on a stereographic net such as a Wulff net or Lambert net. The pole to each face is plotted on the net. Each point is labeled with its Miller index. The final plot allows the symmetry of the crystal to be established.

Crystallographic methods now depend on the analysis of the diffraction patterns of a sample targeted by a beam of some type. X-rays are most commonly used; other beams used include electrons or neutrons. Crystallographers often explicitly state the kind of beam used, as in the terms X-ray crystallography, neutron diffraction, and electron diffraction. These three types of radiation interact with the specimen in different ways. X-rays interact with the spatial distribution of electrons in the sample. Electrons are charged particles and therefore interact with the total charge distribution of both the atomic nuclei and the electrons of the model. Neutrons are scattered by the atomic nuclei through the strong nuclear forces, but in addition, the magnetic moment of neutrons is non-zero. They are therefore also scattered by magnetic fields.

## **2.6.2 Relation Between Crystallography and Vector space**

In chemistry, crystallography is the study of the arrangement of atoms inside crystalline crystals. It gives extensive information on crystals' three-dimensional structure, which is necessary for understanding the characteristics and behavior of diverse materials. In the following ways, crystallography is connected to vector spaces:

**Lattice Vectors:** In crystallography, the arrangement of atoms in a crystal is described in terms of a crystal lattice. A crystal lattice is a periodic arrangement of points in three-dimensional space. The lattice vectors, also known as basis vectors, define the translation symmetries of the crystal. These lattice vectors form a vector space, and operations involving lattice vectors (addition, scalar multiplication) are consistent with vector space properties.

**Reciprocal Space:** Reciprocal space can be described as a vector space where each point represents a set of reciprocal lattice vectors. The properties of vectors in this space are essential for understanding diffraction patterns and other crystallographic techniques.

**Structure Factor:** The structure factor in crystallography is a complex number associated with each set of lattice planes and describes how the crystal scatters X-rays or electrons. The structure factor can be thought of as a vector in complex space, and its properties are related to vector space operations.

**Crystallographic Groups:** Crystallographic groups are mathematical groups that explain crystallographic symmetry operations. These operations can be represented as vector space linear transformations. Crystal symmetry elements are intimately connected to vector space qualities.

**Description of Atomic Positions:** The positions of atoms in a crystal lattice are often described using fractional coordinates within a unit cell. These coordinates can be viewed as vectors within the unit cell, and vector space concepts can be applied to manipulate and analyze these coordinates.

In summary, crystallography and vector spaces are closely related in the context of understanding the spatial arrangement and properties of crystalline materials. Vector spaces provide a mathematical framework for describing and analyzing the symmetries, structures, and properties of crystals in three-dimensional space dimensional space.

**2.7 Chemical modeling**



## **2.7.1 What is Chemical modeling?**

Chemical modeling, also known as computational chemistry or molecular modeling, is a field of science that involves using computer simulations and mathematical techniques to study and predict the behavior of molecules and chemical systems. Researchers in the domains of chemistry, biochemistry, and allied sciences can benefit greatly from this technique. Instead of spending money on costly and time-consuming laboratory experiments, scientists can learn more about the interactions, structure, and characteristics of molecules through chemical modeling.

Common techniques and software used in chemical modeling include molecular dynamics simulations, density functional theory (DFT), and various quantum chemistry methods. These techniques rely on mathematical equations and algorithms to simulate the behavior of atoms and molecules, allowing scientists to make predictions and understand complex chemical phenomena.

## **2.7.1 Relation between Chemical Modeling and vector space**

Chemical modeling is a technique used in chemical engineering process design that involves using purpose-built software to define a system of interconnected components, which are then solved so that the steady-state or dynamic behavior of the system can be predicted.

In terms of linear algebra, it is used to study the properties and behavior of molecules and chemical reactions. For example, a vector space can be defined for a set of molecular orbitals, and linear combinations of these orbitals can be used to describe the electronic structure of molecules.

In addition to linear algebra, differential equations, numerical analysis, and optimization are also important mathematical concepts used in chemical modeling. Differential equations are used to describe the behavior of chemical systems over time. Numerical analysis is used to solve these equations, which can be very complex. Optimization is used to find the best solution to a problem, such as the optimal design of a chemical reactor. Here's how these mathematical tools are related to chemical modeling:

**Representation of Molecular Structure:** In chemical modeling, molecules are often represented as sets of atoms and bonds. These representations can be mathematically described using vectors and matrices. For example, a molecular graph can be represented as an adjacency matrix, where each element of the matrix indicates the presence or absence of a bond between two atoms. Linear algebra is used to perform operations on these matrices, such as matrix multiplication and diagonalization, to extract information about the molecule's structure and properties.

**Quantum Mechanics and Wavefunctions:** Wavefunctions are used in quantum chemistry, a subfield of chemical modeling, to characterize the behavior of electrons in atoms and molecules. These wavefunctions are mathematical functions with complicated structures that occur in vector spaces. The Schrödinger equation describes the behavior of electrons in quantum mechanics and may be solved using linear algebra. Several approaches in quantum chemistry, including density functional theory and Hartree-Fock theory, use linear algebra to discover approximations of the solutions to this equation.

**Eigenvalue Issues:** In chemical modeling, eigenvalue issues and diagonalization are often used methods. For example, the molecular energy levels and wavefunctions may be inferred from the eigenvalues and eigenvectors of a molecular Hamiltonian matrix. Usually, linear algebra techniques are used to handle these eigenvalue issues.

**Matrix Operations:** Chemical modeling often involves matrix operations, such as solving systems of linear equations and performing matrix-vector multiplications. These operations are used to calculate molecular properties, solve electronic structure problems, and simulate molecular dynamics. Linear algebra provides the mathematical framework for performing these calculations efficiently.

In summary, chemical modeling in chemistry relies on concepts from vector spaces and linear algebra to represent and manipulate molecular and chemical data. These mathematical tools are essential for solving complex problems in quantum chemistry, molecular modeling, and chemical informatics, allowing scientists to gain insights into molecular structure, properties, and behavior.

# **2.8 Chemical Engineering**



## **2.8.1 What is Chemical Engineering?**

Chemical engineering is a branch of engineering that deals with the design, construction, and operation of machines and plants that perform chemical reactions to solve practical problems or make useful products. It is a field that sits at the intersection of science and technology, and it involves the application of principles from chemistry, physics, biology, mathematics, and economics to solve technical problems. Chemical engineers are responsible for developing economical commercial processes to convert raw materials into useful products.

## **2.8.2 Relation Between Chemical Engineering and Vector Space**

In chemical engineering, vector spaces are used to represent the state of a system and its properties. For example, the state of a chemical reaction can be represented as a vector in a high-dimensional space, where each dimension corresponds to a different chemical species. Vector spaces are also used to represent the properties of materials, such as their thermal conductivity, electrical conductivity, and mechanical properties. This is how these ideas are related:

**Linear Algebra in Chemical Engineering:** Linear algebra is a branch of mathematics that deals with vector spaces, matrices, and linear transformations. It provides tools to analyze and solve systems of linear equations and perform various operations on vectors and matrices.

**Material and Energy Balances:** In chemical engineering, one of the fundamental concepts is material and energy balances. These balances involve tracking the flow of materials and energy in a chemical process. These balances can be represented mathematically using a system of linear equations. The coefficients in these equations can form a matrix, and the variables (concentrations, temperatures, flows, etc.) can be represented as vectors. By solving these linear equations using linear algebra techniques, chemical engineers can determine the composition and behavior of substances within a process.

**Process Modeling and Simulation:** Chemical engineers often use mathematical models to simulate and predict the behavior of chemical processes. These models are typically represented using differential equations. Linear algebra is used to discretize these differential equations, resulting in a system of algebraic equations. These equations can be solved to understand how various parameters and variables change with time in a chemical process.

**Eigenvalues and Eigenvectors:** Eigenvalues and eigenvectors are concepts from linear algebra that are used in chemical engineering to analyze and control processes. For instance, they can be used to study stability and control in dynamic systems or to identify dominant modes of behavior.

**Optimization:** Chemical engineers frequently encounter optimization problems in process design and operation. They aim to find the optimal set of operating conditions or design parameters that maximize efficiency, minimize costs, or meet specific criteria. Linear programming, a mathematical optimization technique, is commonly used to solve linear optimization problems. In this context, vector spaces can represent decision variables, and linear algebra is used to find the optimal solutions within these spaces.

In summary, the relationship between chemical engineering and vector spaces is evident in the mathematical tools and techniques that chemical engineers use to model, analyze, and optimize chemical processes. Linear algebra, as a foundation of vector spaces, plays a crucial role in solving equations, performing simulations, and making informed decisions in chemical engineering applications.

# **2.9 Some Mathematical Problems in Chemistry Based on Vector Space**

**Example: 1**

Balance the following chemical equation. When solutions of Sodium Phosphate and Barium Nitrate are mixed, the result is Barium Phosphate and Sodium Nitrate. The unbalanced equation is:

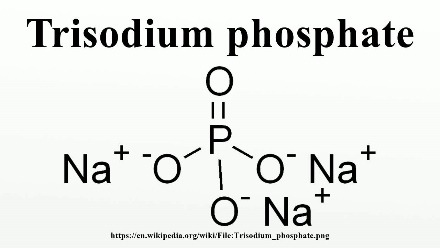
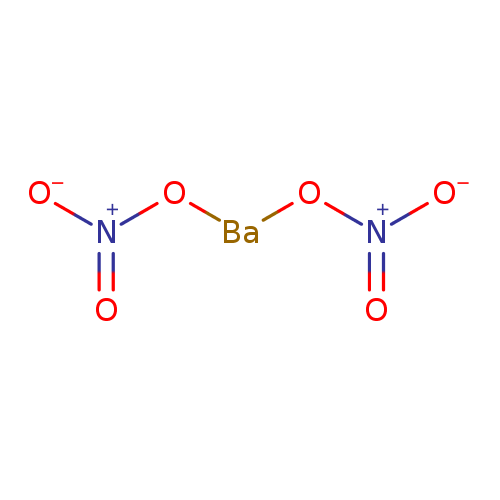
Let the balance equation be:

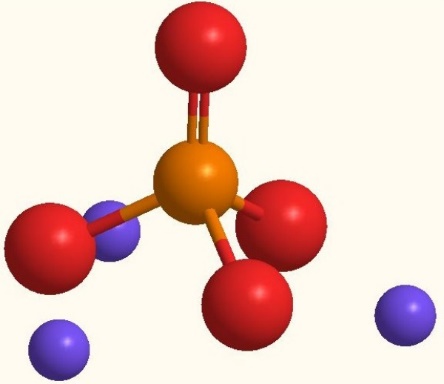
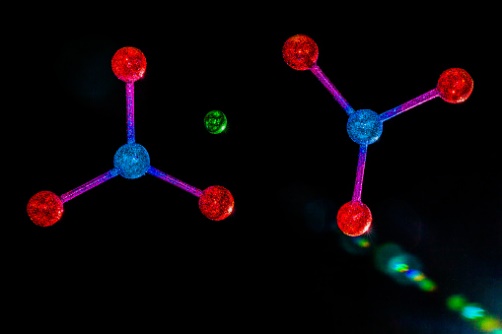
The augmented matrix of the corresponding be:

This means that,

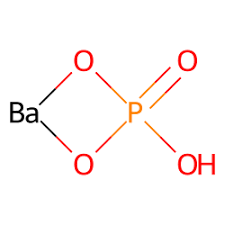
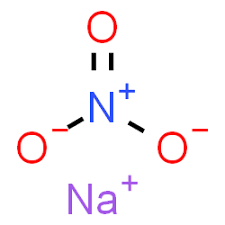
Choose Then

The balanced equation be:

Trisodium Phosphate Barium Naitrate

Barium Phosphate Sodium Naitrate

**Example: 2**

In a laboratory experiment, a researcher collects spectral data from three different samples using UV-Visible spectroscopy. The data for each sample is represented as a vector in a 1D vector space, with each element in the vector corresponding to the intensity of light at a specific wavelength.

Sample 1 Spectrum (in arbitrary units): S₁ = [0.2, 0.5, 0.8, 1.0, 0.9, 0.6]

Sample 2 Spectrum (in arbitrary units): S₂ = [0.1, 0.3, 0.6, 0.9, 0.7, 0.4]

Sample 3 Spectrum (in arbitrary units): S₃ = [0.3, 0.6, 1.0, 1.1, 1.2, 0.9]

The researcher is interested in comparing these samples to see if they share any common spectral features and wants to perform vector space operations to analyze the data.

1. Calculate the dot product between Sample 1 (S₁) and Sample 2 (S₂).

2. Find the magnitude (Euclidean norm) of the vector representing Sample 3 (S₃).

3. Calculate the cosine similarity between Sample 1 (S₁) and Sample 3 (S₃).

Solve:

1. Dot Product: The dot product between two vectors A and B is calculated as: A · B = Σ(Aᵢ \* Bᵢ) for all i, where i ranges over the elements of the vectors.

S₁ · S₂ = (0.2 \* 0.1) + (0.5 \* 0.3) + (0.8 \* 0.6) + (1.0 \* 0.9) + (0.9 \* 0.7) + (0.6 \* 0.4)

S₁ · S₂ = 0.02 + 0.15 + 0.48 + 0.90 + 0.63 + 0.24

S₁ · S₂ = 3.42

2. Magnitude (Euclidean Norm): The magnitude of a vector V is calculated as: ||V|| = √(Σ(Vᵢ²)) for all i, where i ranges over the elements of the vector.

||S₃|| = √((0.3²) + (0.6²) + (1.0²) + (1.1²) + (1.2²) + (0.9²))

||S₃|| = √(0.09 + 0.36 + 1.00 + 1.21 + 1.44 + 0.81)

||S₃|| = √(5.91)

||S₃|| ≈ 2.43 (rounded to two decimal places)

3. Cosine Similarity: The cosine similarity between two vectors A and B is calculated as: cos(θ) = (A · B) / (||A|| \* ||B||)

Cosine Similarity (S₁, S₃) = (S₁ · S₃) / (||S₁|| \* ||S₃||)

Cosine Similarity (S₁, S₃) = (3.42) / (1.38 \* 2.43)

Cosine Similarity (S₁, S₃) ≈ 0.925 (rounded to three decimal places)

So, the answers to the exercises are:

1. The dot product between Sample 1 and Sample 2 is 3.42.

2. The magnitude (Euclidean norm) of the vector representing Sample 3 is approximately 2.43.

3. The cosine similarity between Sample 1 and Sample 3 is approximately 0.925.

# **Chapter Three**

# **Computer Science in Vector Space**



# **3.1 What is Computer Science:**

Computing is part of everything we do. Computing drives innovation in engineering, business, entertainment, education, and the sciences—and it provides solutions to complex, challenging problems of all kinds.

Computer science is the study of computers and computational systems. It is a broad field which includes everything from the algorithms that make up software to how software interacts with hardware to how well software is developed and designed. Computer scientists use various mathematical algorithms, coding procedures, and their expert programming skills to study computer processes and develop new software and systems.

# **3.2 Application of Vector Space in Computer Science**

Vector space models are a popular application of computer science in the field of information retrieval systems. Vector space models are used to consider the relationship between data that are represented by vectors. This allows us to compare the similarity of two vectors from a geometric perspective. In computer science, vector space models have many applications, including:

**Natural Language Processing (NLP):** In NLP, text data is often represented as sequences of words. Word embedding use vector spaces to map each word to a vector, capturing semantic relationships. For example, in Word2Vec, words with similar meanings are closer in the vector space.

**Information Retrieval:** In information retrieval, documents and queries are represented as vector space models. In a high-dimensional space, each document or query is represented as a vector whose dimensions are related to the frequency of words in the document or query. Cosine similarity is used to rank documents based on their similarity to a query.

**Computer Graphics:** In 3D computer graphics, objects are represented in a 3D Cartesian coordinate system. Points and vectors in this space describe the position, orientation, and scale of objects. Transformations are applied in this vector space to render 3D scenes.

**Image Processing:** Images are represented as matrices of pixel values. These matrices can be treated as vectors in a high-dimensional space. Image processing operations, like convolution and filtering, manipulate these vector representations.

**Data Compression:** Vector quantization techniques use vector spaces to represent data. The process involves encoding data points using vectors from a codebook, effectively compressing data. Vector space models can be used to compress data by representing it in a lower-dimensional space without losing much information.

**Graph Theory:** Graph data can be represented as matrices or tensors, where nodes are mapped to vectors in an embedding space. These embeddings are learned by algorithms such as node2vec and graph neural networks to carry out tasks like link prediction and node classification.

**Quantum Computing:** Quantum bits or qubits are represented as vectors in a complex vector space. Quantum algorithms manipulate these vectors to perform computations, exploiting properties like superposition and entanglement for tasks like factoring large numbers and simulating quantum systems.

In each of these applications, vector spaces provide a structured way to represent and manipulate data, allowing for various algorithms and techniques to be applied effectively in computer science and related fields.

# **3.3 Natural Language Processing**



## **3.3.1 What is Natural Language Processing (NLP)**

Natural Language Processing (NLP) is a field of computer science that deals with the interaction between computers and human language. It is a subfield of Artificial Intelligence (AI) that focuses on enabling computers to understand, interpret, and manipulate human language in a way that is similar to how humans do.

NLP combines computational linguistics, machine learning, and deep learning models to process human language in the form of text or voice data and to 'understand' its full meaning, complete with the speaker or writer's intent and sentiment. NLP drives computer programs that translate text from one language to another, respond to spoken commands, and summarize large volumes of text rapidly—even in real time.

Some of the NLP tasks include speech recognition, part-of-speech tagging, word sense disambiguation, named entity recognition, sentiment analysis, and machine translation. These tasks break down human text and voice data in ways that help the computer make sense of what it's ingesting.

NLP has a wide range of applications in various fields such as healthcare, finance, customer service, education, and more. For instance, NLP can be used to extract information from medical records or clinical notes to improve patient care. It can also be used to analyze customer feedback on social media platforms to improve customer service.

## **3.3.2 Relation between Natural Language Processing (NLP) and vector space**

Natural Language Processing (NLP) in computer science is closely related to vector spaces and linear algebra. These mathematical concepts and techniques play a crucial role in representing and processing natural language data. Here's an overview of the relationship between NLP, vector spaces, and linear algebra:

Vector Space Models (VSM): In NLP, vector space models are a popular method for representing textual data. These models represent words, phrases, or documents as vectors in a high-dimensional space. Each dimension of this space corresponds to a unique term or word. The presence or absence of a word in a document is encoded as the value of the corresponding dimension in the vector. This allows for efficient mathematical operations and comparisons between documents.

**Word Embeddings:** Word embeddings are a popular application of vector spaces and linear algebra in NLP. Word2Vec, GloVe, and FastText are examples of techniques used to create word embeddings. These techniques map words from a vocabulary into dense, continuous-valued vectors in a vector space. These embeddings capture semantic and syntactic relationships between words, making them useful for various NLP tasks, such as text classification, sentiment analysis, and machine translation.

**Document Representation:** Vector space models can be extended to represent entire documents, going beyond word-level embeddings. Documents are encoded in vector space using document-term matrices or TF-IDF (Term Frequency-Inverse Document Frequency) representations, allowing for a variety of text mining and information retrieval tasks.

**Similarity and Distance Measures:** Linear algebra is used to calculate similarity and distance measures between vectors representing words or documents. Common distance metrics include Euclidean distance, cosine similarity, Jaccard similarity, and more. These measures help in tasks like document clustering, information retrieval, and recommendation systems.

**Text Classification:** Linear algebra is used in text classification tasks, where vectors representing documents are mapped to specific categories or labels using linear classifiers like logistic regression or support vector machines.

**Graph Theory:** Another mathematical concept that is used in NLP is graph theory. Graph theory is used to model relationships between words in a sentence or document. This is useful for tasks such as named entity recognition, where we want to identify entities such as people, organizations, and locations in a text.

# **3.4 Information Retrieval (IR)**



## **3.4.1 What is Information Retrieval (IR)**

Information Retrieval (IR) is a field of computer science that deals with the organization, storage, retrieval, and evaluation of information from document repositories, particularly textual information. It is the activity of obtaining material that satisfies an information need from within large collections of documents stored on computers.

IR is widely used in search engines to help users find relevant information based on their queries. The process involves searching for documents that contain the user's query terms, ranking the documents based on their relevance to the query, and presenting the results to the user.

An IR system has the ability to represent, store, organize, and access information items. It searches over billions of documents stored on millions of computers. The system assists users in finding the information they require but it does not explicitly return the answers to the question. It notifies regarding the existence and location of documents that might consist of the required information.

The IR system extends support to users in browsing or filtering document collection or processing a set of retrieved documents. The system can also be used for spam filtering by providing manual or automatic means for classifying emails so that they can be placed directly into particular folders.

## **3.4.2 Relation between Information Retrieval and vector space**

Information retrieval (IR) in computer science often makes use of the vector space model (VSM) as a key framework for representing and comparing documents and queries. The VSM is a mathematical model that helps organize and retrieve information, making it a fundamental concept in information retrieval. Here's the relationship between information retrieval and the vector space model.

Similarity Measurement: Once documents and queries are transformed into vector representations, the vector space model allows for the calculation of similarity between a query vector and document vectors. Cosine similarity is a commonly used measure that calculates the angle between vectors and provides a quantitative measure of how similar a document is to a query.

Scalability: The vector space model is designed to efficiently retrieve documents from large collections. Indexing and ranking based on vector representations can be carried out quickly, which is essential for applications such as web search engines that deal with massive document repositories.

Term Frequency-Inverse Document Frequency (TF-IDF): The VSM often employs the TF-IDF weighting scheme to assign values to elements in the vectors. TF-IDF is a statistical measure that considers the term frequency within a document and the inverse document frequency across the entire collection. This helps emphasize terms that are more informative and less common across the document collection.

In conclusion, the vector space model's flexibility and adaptability make it a powerful tool for information retrieval. It allows for the efficient processing and retrieval of relevant information from extensive document collections, and it can be customized and tuned to suit different retrieval requirements. This model is foundational in the field of information retrieval and is widely applied in search engines, recommendation systems, and document retrieval systems.

# **3.5 Computer Graphics**



## **3.5.1 What is Computer Graphics?**

Computer graphics in computer science is a field that deals with the creation, manipulation, and representation of visual images and animations using computers. It encompasses a wide range of topics and techniques aimed at generating, processing, and displaying visual content. Computer graphics is essential in various applications, including video games, simulations, computer-aided design (CAD), virtual reality, scientific visualization, and more.

Computer graphics can be interactive or non-interactive. Interactive computer graphics allow users to tell a computer how to generate an image, while non-interactive computer graphics do not allow users to determine how images are generated.

## **3.5.2 Relation Between Computer Graphics and Vector Space**

In computer graphics, vectors are used to represent geometric objects such as points, lines, and planes in 2D and 3D space. Vectors are also used to represent the attributes of vertices in 3D models.

A vector space in linear algebra is a group of vectors that may be multiplied and added to by scalars. The mathematical procedures utilized in computer graphics are defined by the features of vector spaces. Here's an exploration of their relationship:

**Representation of Geometric Data:** In computer graphics, geometric objects like points, lines, and polygons are often represented using vectors. A vector in a vector space can represent a point in 2D or 3D space, making it suitable for storing the coordinates of vertices in a 3D model. For instance, a 2D point (x, y) can be represented as a 2D vector (x, y) in a two-dimensional vector space.

**Matrices and Transformations:** A key idea in vector spaces, matrices are widely employed in computer graphics to carry out transformations including translation, rotation, scaling, and projection. Vectors can be transformed to change an object's size, orientation, and location within a graphical scene.

**Linear Algebra:** Vector spaces and matrices are part of linear algebra, which is a fundamental body of mathematics that underpins computer graphics. It is used to solve systems of linear equations, perform transformations, calculate lighting and shading effects, and more. Operations in vector spaces, such as addition and scalar multiplication, are used to manipulate graphical elements efficiently.

**Color Representations:** Colors in computer graphics are often represented as vectors in color spaces. For example, the RGB color model represents colors as vectors in a three-dimensional color space. Vectors are used to perform color blending and transformations.

**Homogeneous Coordinates:** Homogeneous coordinates, a concept derived from vector spaces, are used in computer graphics to represent 3D transformations, including translations, as matrix multiplications. This allows for the efficient representation of both translation and non-translation transformations as matrix operations.

**Shading and Lighting:** In shading and lighting computations, vectors are utilized to indicate the direction of light sources and surface normal. The dot product between these vectors is used to compute lighting effects.

**Vector Graphics Formats:** Vector graphics file formats, like SVG (Scalable Vector Graphics), store graphical data using vector representations. This allows for the scalable rendering of graphics without loss of quality.

In conclusion, many facets of computer graphics are supported mathematically by vector spaces. They are used to represent geometric data, perform transformations, and handle color information. Computer graphics applications rely heavily on the concepts of linear algebra in vector spaces for the creation and manipulation of graphical material.

# **3.6 Image Processing**



## **3.6.1 What is Image Processing?**

Image processing is the process of manipulating digital images using computer algorithms. It is an essential preprocessing step in many applications, such as face recognition, object detection, and image compression. The process involves several phases such as image enhancement, restoration, and segmentation.

In image enhancement, the goal is to improve the quality of an image by increasing its contrast, brightness, or sharpness. Image restoration aims to remove noise or blur from an image. Image segmentation is the process of dividing an image into multiple segments or regions based on their characteristics.

Deep learning has revolutionized the world of computer vision—the ability for machines to “see” and interpret the world around them. In particular, Convolutional Neural Networks (CNNs) were designed to process image data more efficiently than traditional Multi-Layer Perceptron’s (MLP). Since images contain a consistent pattern spanning several pixels, processing them one pixel at a time—as MLPs do—is inefficient. This is why CNNs that process images in patches or windows are now the de-facto choice for image processing tasks. Image processing in computer science involves the use of software and computational methods to perform various operations on images, such as enhancement, analysis, recognition, and transformation.

## **3.6.2 Relation Between Image Processing and Vector Space**

Image processing and vector spaces are related in the context of representing and manipulating images using mathematical methods. In image processing, images are often treated as two-dimensional arrays of pixels, and these pixels can be viewed as vectors in a multi-dimensional vector space. Here's how image processing and vector spaces are related:

**Vector Representation of Images:** In digital image processing, images are typically represented as a grid of pixels. Each pixel's color or intensity values can be considered as components of a vector. For grayscale images, each pixel corresponds to a single scalar value, while for color images, each pixel is represented as a vector with multiple components (e.g., RGB values). Thus, you can think of an image as a collection of vectors, where each vector represents a pixel in the image.

**Digital Image Representation:** Images are represented as a grid of pixels, with each pixel containing color or intensity information. In computer science, digital images are typically stored as arrays of numerical values (e.g., in the form of matrices or tensors).

**Distance Metrics:** Similarity or dissimilarity between images can be measured using distance metrics in vector spaces. For example, the Euclidean distance can be used to compare the similarity of pixel values between two images. This is useful in tasks like image retrieval and content-based image retrieval.

**Vector Quantization:** In image compression, vector quantization techniques like the k-means algorithm are used to group similar image patches into clusters, effectively encoding them as vectors in a reduced-dimensional space. This reduces the amount of data required to represent the image.

**Vector Spaces for Color Models:** Color spaces, such as RGB, CMYK, and LAB, are often represented as vector spaces, with color values treated as components of vectors. Transformations between these color spaces are mathematically represented as linear transformations in vector spaces.

**Superpixels and Image Segmentation:** Image segmentation techniques may involve dividing images into regions or superpixels, which can be represented as collections of vectors. This representation simplifies the analysis of regions within the image.

Overall, the relationship between image processing and vector spaces is evident in the way images are mathematically represented, transformed, and analyzed using techniques from linear algebra and vector space mathematics. These representations and operations are fundamental in the field of image processing, providing a mathematical framework for various image analysis tasks.

# **3.7 Data Compression**



## **3.7.1 What is Data Compression?**

In computer science, data compression is the process of modifying, encoding or converting the bits structure of data in such a way that it consumes less space on disk. Compression reduces the cost of storage, increases the speed of algorithms, and reduces the transmission cost. Compression is achieved by removing redundancy, that is repetition of unnecessary data. Coding redundancy refers to the redundant data caused due to suboptimal coding techniques.

Data compression can be expressed as a decrease in the number of bits required to illustrate data. One important area of research is data compression. It deals with the art and science of storing information in a compact form.

There are two types of data compression: lossless and lossy. In lossless compression, the compressed data can be fully reconstructed to its original form without any loss of information. This type of compression is essential for preserving data integrity and is commonly used for text files, program code, and any data where loss of information is unacceptable. Popular algorithms for lossless compression include:

* Run-Length Encoding (RLE)
* Huffman coding
* Lempel-Ziv-Welch (LZW)
* DEFLATE (used in ZIP files)

Lossy compression reduces the size of data by removing some of the less critical information. While this results in a smaller file size, it also leads to some loss of data quality. Lossy compression is commonly used in multimedia applications, such as images, audio, and video. Popular algorithms for lossy compression include:

* JPEG (for images)
* MP3 (for audio)
* H.264 (for video)

Without data compression, a 3-minute song would be over 100Mb in size, while a 10-minute video would be over 1Gb in size.

## **3.7.2 Relation Between Data Compression and vector space**

The relation between data compression in computer science and vector spaces lies in the fact that data, whether it's text, images, audio, or other types of information, can often be represented in a mathematical form as vectors in a multi-dimensional space. This representation can be leveraged in various compression techniques to reduce the size of the data. Let's explore this relationship in more detail:

**Transform Coding:** Transform coding is a compression technique that transforms the data into a different basis, typically a set of orthogonal basis functions. The transformed data can often be represented as vectors in a new space, where some coefficients can be discarded or quantized to achieve compression. For example, the Discrete Cosine Transform (DCT) is used in JPEG compression for images and MPEG compression for video.

**Sparse Coding:** In sparse coding, data is represented as a linear combination of basis vectors. These basis vectors can be considered as vectors in a vector space. By selecting a sparse set of coefficients in this space, you can achieve compression while still approximating the original data.

**Vector Space Models in Text Compression:** In text compression, documents can be represented as vectors in a vector space model, where each term in the document corresponds to a dimension. Various text compression techniques, such as term frequency-inverse document frequency (TF-IDF) and Latent Semantic Analysis (LSA), utilize these vector space representations to reduce the dimensionality and compress textual data.

In summary, the relationship between data compression in computer science and vector spaces arises from the use of mathematical representations of data as vectors in multi-dimensional spaces. Compression techniques often exploit properties of these vector spaces to reduce data size while attempting to preserve essential information. This relationship highlights the intersection between linear algebra, signal processing, and data compression in the field of computer science.

# **3.8 Mathematical Problem and Solution**

Problem:  
Let’s consider a set of vectors in such that , and. We want to determine whether the vector belongs to the span of .

Solution:  
To solve this problem, we need to find the coefficients , , and such that. This can be written as a system of linear equations:

We can write this system in matrix form as, where:

We can solve this system using Gaussian elimination or any other method of our choice. Here are the steps for solving this system using Gaussian elimination:

1. The augmented matrix :

2. Subtract times the first row from the second row:

3. Subtract 3 times the first row from the third row:

4. Multiply the second row by -2/5:

5. Add 3 times the second row to the third row:

6. Multiply the third row by -1/2:

7. Add 1 times the third row to the second row:

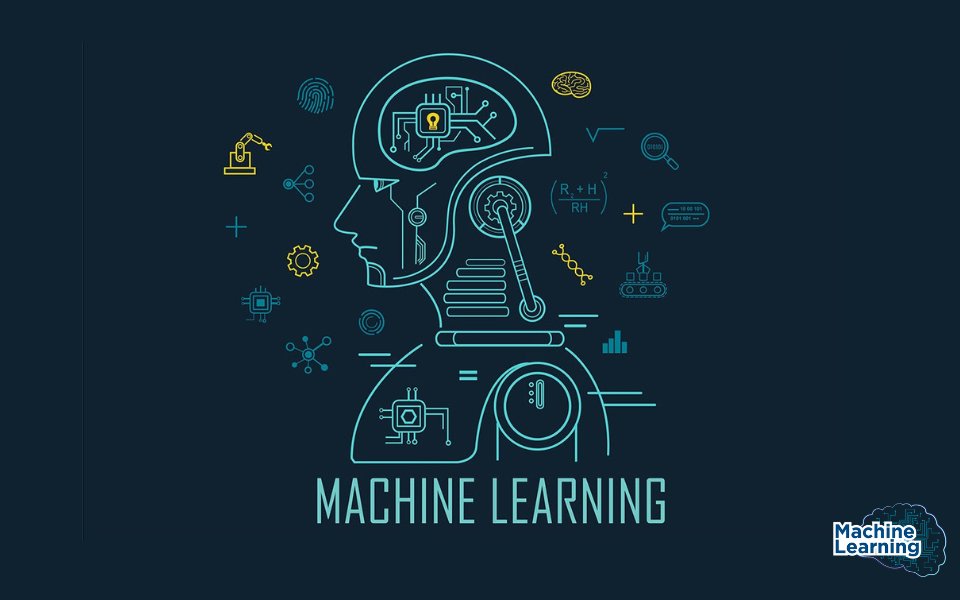
8. Add 1 times the second row to the first row:

9. Add 1 times the third row to the first row:

Therefore, the solution to the system of linear equations is:

# **Chapter Four**

# **Machine Learning in Vector Space**



# **4.1 What is Machine Learning?**

Machine learning is a subfield of artificial intelligence (AI) that focuses on the development of algorithms and statistical models that enable computer systems to learn and make predictions or decisions without being explicitly programmed. In traditional programming, humans write explicit instructions for a computer to follow, but in machine learning, the computer learns from data and experiences to improve its performance on a specific task.

Machine learning algorithms are designed to identify patterns, extract insights, and make predictions or decisions based on data. They can be broadly categorized into three main types:

Supervised Learning: In supervised learning, the algorithm is trained on a labeled dataset, which means that the input data is paired with corresponding target outputs. The algorithm learns to map input data to the correct outputs and can make predictions on new, unseen data.

Unsupervised Learning: Unsupervised learning deals with unlabeled data. The algorithm tries to find patterns or structures within the data, such as clustering similar data points together or reducing the dimensionality of the data.

Reinforcement Learning: In reinforcement learning, an agent interacts with an environment and learns to make a sequence of decisions to maximize a reward signal. This type of learning is often used in scenarios where an agent needs to make a series of actions to achieve a goal, such as in robotics and game playing.

# **4.2 Application of Machine Leaning in Vector Space**

In vector space, machine learning offers a wide range of applications. Kernel-based unsupervised and supervised learning methods are one of the most popular applications. The standard Euclidean inner product is substituted with more complex kernel-induced inner products associated with the relevant kernel-induced vector spaces in this technique. This method delivers more accurate similarity measurements for any pair of items.

**Word Embeddings in Natural Language Processing:**

* Word2Vec: Word2Vec is a common word embedding approach that converts words into continuous vector representations. It learns to group words with similar contexts together in vector space. This is accomplished using either the Continuous Bag of Words (CBOW) or the Skip-gram models.
* GloVe (Global Vectors for Word Representation): GloVe is another word embedding approach that creates word vectors by leveraging global word co-occurrence information. It expresses words in such a way that their semantic links are captured.
* FastText expands word embeddings to the subword level. terms are represented as the sum of their character n-grams, allowing it to handle out-of-vocabulary terms and capture morphological information.

**NLP Document Classification:**

* TF-IDF Vectorization: A typical approach for vectorizing text texts is Term Frequency-Inverse Document Frequency (TF-IDF). Documents are represented as vectors, with each dimension corresponding to a unique word and the values indicating the significance of that word in the text.
* Approaches Based on Word Embedding: Whole documents may be represented with word embeddings. To construct a document vector for purposes like as sentiment analysis or classification, you can average or sum the word vectors in the document.

**Image Embeddings in Image Processing:** Convolutional Neural Networks (CNNs) are used to extract image features that are then mapped into a vector space. These feature vectors can be used for tasks like image classification, object detection, and face recognition.

Siamese networks learn embeddings for pairs of images in such a way that similar images are close in the embedding space, making them useful for image similarity tasks.

**Collaborative Filtering:** In collaborative filtering, users and items are represented as vectors in a latent space. Machine learning models are used to map users and items to these vectors, and recommendations are made based on the similarity between user and item vectors.

**Anomaly Detection:** One-Class Support Vector Machines (SVM) learn to separate normal data points from anomalies in a high-dimensional space, effectively detecting outliers. Autoencoders are neural networks that may be used to identify anomalies by learning to rebuild input data. Anomalies cause more reconstruction mistakes.

**Embeddings of graphs:** Graph neural networks (GNNs) are used to learn graph embeddings for nodes. By examining the vector space interactions between nodes in the network, these embeddings may be used for tasks such as node categorization, link prediction, and community discovery.

In all of these applications, the key is to represent data in vector form, where each dimension of the vector captures certain information or features, and then use machine learning techniques to process, analyze, and make predictions or decisions based on the relationships and patterns found in the vector space.

# **4.3 Word Embedding**

## **4.3.1 What is Word Embedding**

Word embeddings are a type of word representation that allows words with similar meanings to have a similar representation. They are a distributed representation of text that is perhaps one of the key breakthroughs for the impressive performance of deep learning methods on challenging natural language processing problems.

## **4.3.2 Relation Between Word Embedding and Vector Space**

Word embeddings are learned representations of words in a vector space that capture semantic and syntactic information in machine learning. The vector space is selected so that semantically comparable phrases are mapped to close places in the space. The relationship between word embeddings and vector spaces can be understood as follows:

Vector Space Representation: Word embeddings transform words from their original discrete and symbolic form into a continuous vector space. Each word is mapped to a high-dimensional vector where each dimension of the vector corresponds to a certain aspect or feature of the word's meaning.

**Word Similarity:** In the vector space created by word embeddings, words with similar meanings or contexts are located close to each other. This proximity in the vector space reflects the semantic similarity between words. For example, the vectors for "king" and "queen" are close, indicating their semantic relationship.

**Vector Operations:** Word embeddings provide useful vector operations. For example, if you remove the vector for "man" from the vector for "king" and add the vector for "woman," you obtain a vector that is near to the vector for "queen." This displays word embeddings' capacity to capture semantic links and analogies.

**Continuous Representation:** Unlike traditional one-hot encoding, where words are represented by binary vectors with a single "1" and the rest "0s," word embeddings provide a continuous and distributed representation. This continuous nature captures nuances of word meaning, making it more suitable for machine learning models.

**Transfer Learning:** Word embeddings that have been trained on large text corpora, such as Word2Vec, GloVe, and FastText, may be transferred and utilized as features in various NLP applications. Models can then use the knowledge embodied in these pre-trained vectors.

**Embedding Space:** The vector space in which word embeddings reside is often referred to as the "embedding space." This space is continuous and can have hundreds or even thousands of dimensions, depending on the specific embedding model used.

In conclusion, word embeddings in machine learning function as a link between natural language and vector spaces. They enable the encoding of words in a continuous and semantically relevant vector space, which is required for many NLP tasks and enables machine learning models to interact with and understand textual input. The link between word embeddings and vector spaces is crucial to their success in improving natural language interpretation and processing.

# **4.4 NLP Document Classification**



## **4.4.1 What is NLP Document Classification?**

In Machine Learning, Document Classification is a process of assigning a document to one or more predefined categories based on its content. Vector space models are one of the most popular approaches for representing text data in NLP. In this approach, each document is represented as a vector in a high-dimensional space, where each dimension represents a feature of the document.

NLP document classification in machine learning based on vector space leverages the power of vector representations to capture the textual content and semantic meaning of documents. By converting text data into numerical vectors and employing machine learning algorithms, this approach can automate the process of sorting and categorizing large volumes of textual information. It is widely used in applications like email classification, sentiment analysis, topic modeling, spam detection, and content recommendation.

## **4.4.2 Relation Between NLP (Natural Language Processing) document classification and Vector Space**

The relationship between NLP (Natural Language Processing) document classification in machine learning and vector space is a fundamental one, and it's central to how NLP document classification tasks are approached. Here's a detailed explanation of this relationship:

Document Representation in Vector Space: Text documents are converted from their original textual format into numerical vectors in NLP document categorization. In a high-dimensional vector space, each document is represented as a vector. The vector's dimensions indicate numerous properties or attributes of the document.

**Feature Extraction:** The process of converting text documents into vectors involves feature extraction. Common techniques include Bag of Words (BoW), TF-IDF (Term Frequency-Inverse Document Frequency), and word embeddings. With BoW and TF-IDF, each dimension in the vector corresponds to a word or term, and the value in that dimension represents the frequency, importance, or weight of that word in the document. Word embeddings represent words as dense vectors in a continuous vector space, and document vectors are obtained by aggregating word embeddings.

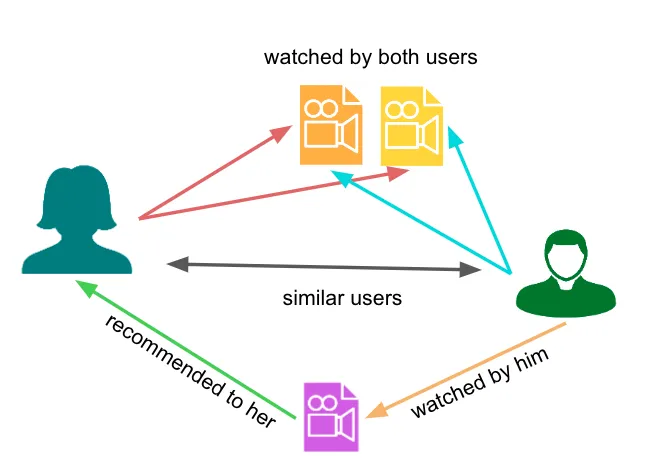
**Semantic Meaning in Vector Space:** In the vector space, documents are positioned relative to each other based on the similarity or dissimilarity of their content. Similar documents are closer to each other in vector space, while dissimilar documents are farther apart. This similarity reflects the semantic meaning of the documents.

**Similarity Measures:** In document classification, similarity measures such as cosine similarity or Euclidean distance are often used in the vector space. Cosine similarity measures the cosine of the angle between two vectors, providing a measure of their similarity. Documents with similar content have a smaller angle and a higher cosine similarity.

**Evaluation in Vector Space:** Model performance in NLP document classification is evaluated based on how well documents are classified in the vector space. Evaluation metrics like accuracy, precision, recall, and F1 score assess how accurately the model assigns documents to their true categories.

In summary, machine learning NLP document categorization is based on translating text documents into vector representations, where the vector space maintains the semantic meaning and relationships between documents. Machine learning methods identify documents in this vector space based on their locations and closeness to category vectors. The representation of documents as vectors in a continuous space is a critical component for automatic text data classification.

# **4.5 Collaborative filtering**

****

## **4.5.1 What is Collaborative filtering?**

Collaborative Filtering is a well-known Machine Learning approach for making suggestions to users based on their interests and preferences. It is a sort of recommender system that makes recommendations based on similarities between users and/or goods. The algorithm's fundamental premise is that people with comparable interests have similar preferences.

In Collaborative Filtering, we tend to find similar users and recommend what similar users like. In this type of recommendation system, we don’t use the features of the item to recommend it, rather we classify the users into clusters of similar types and recommend each user according to the preference of its cluster.

## **4.5.2. Relation Between Collaborative Filtering and Vector Space**

Collaborative filtering in machine learning and vector space have a significant relationship when it comes to recommendation systems. Collaborative filtering is a technique used for making automatic predictions (recommendations) about a user's interests by collecting preferences or behavior information from many users. Vector spaces play a crucial role in the representation of users and items, forming the basis of collaborative filtering. Here's how these concepts are related:

**User and Item Embeddings:** In collaborative filtering, users and items (e.g., products in e-commerce or movies in a streaming service) are represented as vectors in a shared vector space. These vectors are often referred to as embeddings, where each dimension in the vector corresponds to a feature or attribute related to users or items.

**Similarity in Vector Space:** The similarity between users and items is calculated based on their respective vectors in the vector space. The similarity measure, often cosine similarity or dot product, quantifies the relatedness between users and items based on their vector representations.

**Rating Prediction:** Collaborative filtering models predict user ratings or preferences for items by considering the interactions between users and items in the vector space. The predictions are made by identifying users whose preferences are similar to the target user and items that are similar to the target item in the vector space.

**User-Item Recommendations:** Recommendations are generated by selecting items that are most similar to the ones the user has interacted with or rated positively. These recommendations are derived from the user's interactions with items and the relationships between users and items in the vector space.

**Matrix Factorization:** Matrix factorization is a common technique in collaborative filtering where the user-item interaction matrix is factored into two lower-dimensional matrices: one for users and one for items. These factorized matrices serve as the embeddings in the shared vector space.

**Hybrid Models:** Collaborative filtering can be combined with content-based methods, where user and item content features are transformed into vector space representations. Hybrid models use these representations to provide recommendations that consider both user behavior and content attributes.

In summary, collaborative filtering leverages the concept of vector space to represent users and items as vectors to understand user preferences and provide recommendations. Vector space representations enable collaborative filtering models to measure similarity, make predictions, and generate recommendations based on the relationships between users and items in this shared space. This relationship forms the foundation of recommendation systems commonly used in e-commerce, content streaming, and various other domains.

# **4.6 Anomaly Detection**



## **4.6.1 What is Anomaly Detection?**

Anomaly detection is a technique used in machine learning to identify rare events or observations that are statistically different from the rest of the data. It is used to detect unusual behavior that can indicate a problem, such as credit card fraud, server failure, or cyber-attacks. Anomalies can be categorized into three types: point anomaly, contextual anomaly, and collective anomaly.

Machine learning algorithms can be used to detect anomalies in two ways: supervised and unsupervised. Supervised anomaly detection requires a labeled dataset containing both normal and anomalous samples to construct a predictive model to classify future data points. Unsupervised anomaly detection does not require any training data and instead assumes that only a small percentage of data is anomalous and any anomaly is statistically different from the normal samples. Based on these assumptions, the data is then clustered using a similarity measure and the data points which are far off from the cluster are considered to be anomalies.

## **4.6.2 Relation Between Anomaly Detection and Vector Space**

Anomaly detection in machine learning is often closely related to vector spaces because data points are typically represented as vectors in a feature space, and the relationships and distances between these vectors are fundamental to the anomaly detection process. Here's how anomaly detection in machine learning is related to vector spaces:

**Feature Space Representation:** Anomaly detection typically deals with data points in a feature space, which can be thought of as a vector space. Each data point is represented as a vector, with each dimension representing a different feature or attribute of the data. For example, in a fraud detection system, each credit card transaction might be represented as a vector in a high-dimensional space, with features such as transaction amount, location, time, etc., making up the dimensions of the vector.

**Distance Measures:** To identify anomalies, you often measure the distances between data points. In a vector space, this is commonly done using distance metrics such as Euclidean distance, Mahalanobis distance, or cosine similarity. Anomalies tend to be data points that are significantly distant from the central or typical cluster of data points in the vector space. If the distance between a data point and its neighbors is unusually large, it may be considered an anomaly.

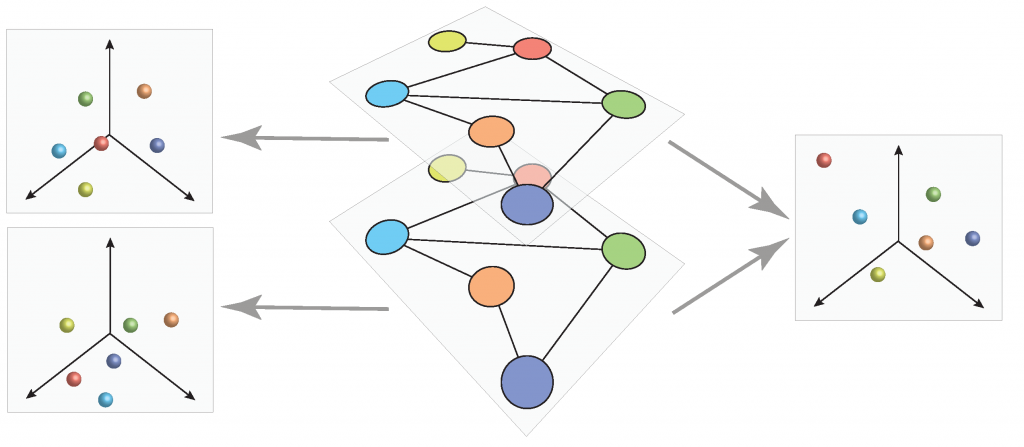
**Density Estimation:** Some anomaly detection methods focus on density estimation within the vector space. They estimate the density of data points, and anomalies are identified as data points residing in regions with much lower density compared to the majority of data. For example, DBSCAN is a density-based clustering algorithm that identifies anomalies as noise points not associated with any dense cluster in the vector space.

**Feature Engineering:** Vector spaces allow for feature engineering, which is necessary for effective anomaly detection. Feature engineering is developing new features, modifying existing ones, or choosing the most relevant features to improve the vector space representation of data. This can help to distinguish abnormalities and make them simpler to spot.

**Model Training**: Machine learning-based anomaly detection models, such as Isolation Forests, One-Class SVM, or autoencoders, operate on the vector representations of data points. These models are trained to distinguish between normal and anomalous data points based on their positions in the vector space. During training, the model learns the characteristics of normal data, allowing it to identify deviations as anomalies during testing.

In essence, the vector space representation of data is fundamental to anomaly detection, as it enables the application of distance metrics, clustering, density estimation, dimensionality reduction, and various machine learning techniques that help identify unusual patterns or data points within the dataset. The specific choice of techniques and methods may vary depending on the nature of the data and the anomaly detection task.

# **4.7 Embeddings of Graphs**



## **4.7.1 What is Graph Embeddings?**

Graph embeddings in machine learning refer to techniques that transform graphs or nodes within a graph into numerical representations that can be used as input for various machine learning models. Embeddings are a way of encoding the structure of a graph into a vector space, which can then be used as input to machine learning models. There are several methods for generating graph embeddings, including matrix factorization, node sequence methods, and deep learning-based methods. Holographic embeddings (HolE) is one such method that learns compositional vector space representations of entire knowledge graphs. These embeddings can be beneficial in a wide range of applications, including recommendation systems, node classification, link prediction, and community discovery. The goal is to capture the structural and relational information present in the graph in a way that is amenable to traditional machine learning algorithms.

## **4.7.2 Relation Between Graph** Embeddings and Vector Space

Graph embeddings in machine learning are closely related to vector spaces because the primary goal of graph embeddings is to represent graph structures or elements (nodes, edges, or entire graphs) as vectors in continuous vector spaces. These vector representations enable the application of various machine-learning techniques that operate on numerical data. Here's how graph embeddings are related to vector spaces:

**Vector Representation:** Graph embeddings convert graph components like nodes, edges, or complete graphs into numerical vectors. In a continuous vector space (e.g., Euclidean space), each element has a relationship with a vector. For example, nodes are represented as points in this space, and their positions capture structural and relational information.

**Feature Space:** Graph embeddings create a feature space where elements of the graph are mapped to vectors. This feature space can be thought of as a high-dimensional vector space. Each dimension of the space corresponds to a feature that encodes information about the graph element. These features are typically learned from the graph's structure and characteristics.

**Mathematical Operations**: Once graph elements are embedded in vector spaces, various mathematical operations can be performed on these embeddings. For instance, you can calculate distances between nodes in the vector space, perform vector arithmetic, or apply machine learning algorithms like clustering, classification, or regression using these vector representations.

**Similarity and Distances:** The computation of similarity between graph components is made possible by vector representations of graph elements. Similarity can be determined using various distance metrics in the vector space, such as Euclidean distance, cosine similarity, or others. This similarity information is valuable for tasks like recommendation, link prediction, and clustering.

**Machine Learning Integration:** Graph embeddings serve as input features for machine learning models. We can use these embeddings as features in traditional machine learning algorithms, deep learning models, or specialized graph-based models. This integration with machine learning allows you to perform various tasks on graph-structured data.

Overall, graph embeddings are a way of encoding the structure of a graph into a vector space. Vector spaces are used to represent the embeddings because they allow for easy mathematical manipulation and comparison of the embeddings. The main goal of graph embedding methods is to pack every node’s properties into a vector with a smaller dimension, hence, node similarity in the original complex irregular spaces can be easily quantified in the embedded vector spaces using standard metrics.

# **4.8 Mathematical Problem and Solution**

Problem:

Given a matrix

find the eigenvalues and eigenvectors.

Solution:

To find the eigenvalues, we solve the characteristic equation , where 𝜆 is the eigenvalue, and 𝐼 is the identity matrix:

So, the eigenvalues are and

To find the eigenvectors, we substitute each eigenvalue into the equation

and solve for .

For :

This leads to .

For :

This leads to .

So, the eigenvalues are and

# **Chapter Five**

# **Robotics in Vector Space**



# **5.1 What is Robotics?**

Robotics is a field of science and engineering that deals with the design, construction, operation, and application of robots. A robot is a programmable machine that can replicate or substitute for human actions. Robotics technology has been applied across industries such as automobile manufacturing, space exploration, and healthcare.

For example, in automobile manufacturing, robots are used to perform tasks such as welding, painting, and assembly of parts. In space exploration, robots are used to explore planets and moons that are difficult or impossible for humans to reach. In healthcare, robots are used to perform surgeries and assist in patient care.

Robotics is a multidisciplinary field that combines mechanical engineering, electrical engineering, computer science, and artificial intelligence. The field is rapidly evolving with new advancements in technology and research.

# **5.2 Relation between Robotics and Vector Space**

In robotics, a vector space is a mathematical framework used to represent the state, configuration, and movement of robotic systems. Vector spaces provide a structured way to describe the position and orientation of robots, the forces and torques acting on them, and the transformations between different coordinate frames. Here are some key concepts related to vector spaces in robotics:

**Configuration Space (C-Space):** The configuration space of a robot is a vector space that represents all possible configurations of the robot. Each point in the configuration space corresponds to a unique robot configuration, where a configuration typically includes the positions and orientations of all its joints. This space can be continuous or discrete, depending on the type of robot.

**Joint Space:** Joint space is a specific type of configuration space that represents the joint angles or joint positions of a robot. In this space, each joint corresponds to a dimension, making it a multi-dimensional vector space.

**Task Space:** Task space, also known as operational space, is a vector space that describes the position and orientation of a robot's end effector or tool in the world. It allows robots to perform tasks and interact with their environment. Transformations are used to convert between joint space and task space.

**Homogeneous Transformations:** Homogeneous transformations are used to describe the transformation between different coordinate frames in robotics. They include both rotation and translation components and are represented as 4x4 matrices. These transformations are crucial for moving between joint space and task space.

**Velocity Space:** Velocity space represents the velocities of a robot's joints or end effector. It is used to describe how a robot moves through its configuration space. The relationship between joint velocities and end effector velocities can be described using the Jacobian matrix.

**Jacobian Matrix:** The Jacobian matrix is a critical concept in robotics, as it relates the velocities of a robot's joints to the linear and angular velocities of its end effector in task space. It helps in solving inverse kinematics problems and controlling the robot's movement.

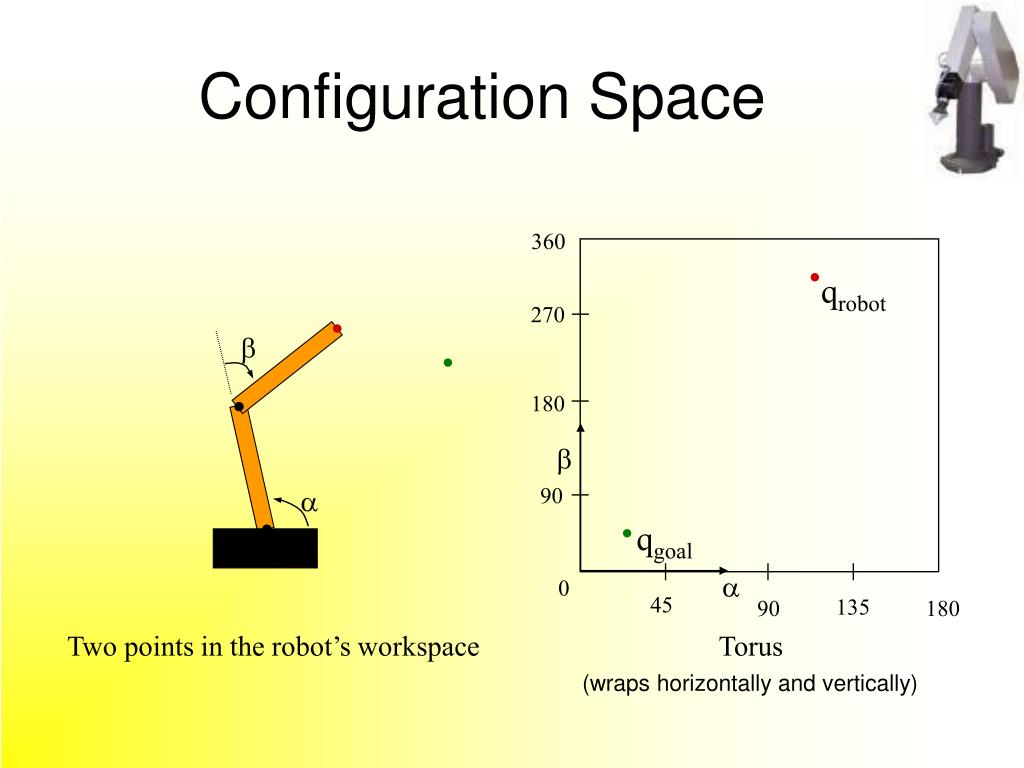
**Lie Groups:** Lie groups are mathematical structures that are used to represent rigid body transformations (rotations and translations). They are essential in robotics for accurately modeling and controlling robot movements.

**State Space:** The state space of a robot represents the complete state of the robot, including its configuration and velocities. It combines both joint and velocity spaces into a single vector space.

**Operational Limits:** Vector spaces in robotics also include operational limits, such as joint limits and workspace boundaries. These limits help ensure safe and effective robot operation.

In summary, vector spaces play a fundamental role in robotics by providing a mathematical framework for modeling, controlling, and planning the movements of robots. They are used to describe robot configurations, movements, and transformations between different coordinate frames, enabling robots to perform various tasks in a structured and precise manner.

# **5.3 Configuration Space (C-Space)?**



## **5.3.1 What is Configuration Space (C-Space)**

Configuration Space (C-Space) is the space of all possible configurations of a robot. A configuration is a complete specification of the position of every point in the system. For example, in a robotic arm, a configuration would specify the angles of each joint. The C-space is a mathematical representation of all possible configurations of a robot.

The C-space is used in motion planning to determine the path that a robot should take to reach its goal. The C-space can be used to represent obstacles in the environment and to determine whether a particular configuration of the robot will result in a collision with an obstacle.

The C-space can be very high-dimensional and complex, making it difficult to visualize and analyze. However, there are techniques for reducing the dimensionality of the C-space and for efficiently searching through it to find feasible paths for the robot.

## **5.3.2 Relation Between Configuration Space and Vector Space**

Configuration Space (C-Space) in robotics is conceptually related to vector spaces, but they serve different purposes and have distinct characteristics. Here are some aspects of this topic:

**C-Space as a Vector Space:** Configuration Space (C-Space) can be thought of as a particular type of vector space. In C-Space, each dimension corresponds to a specific parameter, such as joint angles or positions, that defines the configuration of a robot. These parameters can be considered as elements in a vector space.

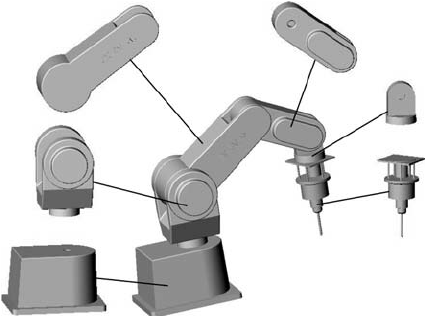
**Dimensionality:** Both C-Space and vector spaces have dimensions. In a C-Space, the dimensionality is determined by the number of degrees of freedom (DOF) of the robot, where each DOF corresponds to a dimension. In a vector space, the dimensionality is determined by the number of independent basis vectors that span the space.

**Coordinate Systems:** C-Space employs coordinates to express a robot's setup. These coordinates, which may contain joint angles, locations, or other pertinent factors, are commonly referred to as generalized coordinates. Coordinates are also used in vector spaces to represent items as linear combinations of basis vectors.

**Mapping:** The mapping between coordinates and physical robot configurations is well-defined in C-Space. Each point in C-Space corresponds to a distinct robot configuration. Vector spaces, on the other hand, map coordinates to points or vectors in the space.

In summary, Configuration Space in robotics is a specialized application of the concept of vector spaces, with a focus on representing the possible configurations of a robot within a given environment. While they share similarities in terms of dimensionality and coordinate systems, their purposes and geometric properties differ significantly.

# **5.4 Joint Space**



## **5.4.1 What is Joint Space?**

In robotics, the term "joint space" refers to the space of all possible joint configurations of a robot. Each joint of a robot has a specific range of motion or set of possible positions, and the combination of all these joint positions defines the robot's overall configuration or posture. The number of dimensions in the joint space is equal to the number of joints in the robot.

Joint space is often used in the context of robot control and motion planning. When a robot needs to perform a task, such as reaching a specific point in its workspace or following a trajectory, the control system needs to compute the appropriate joint angles or joint values to achieve the desired end-effector position or motion.

## **5.4.2 Relation Between Joint Space and Vector Space**

The joint space in robotics and vector space have a relationship in that the joint space can be considered a special kind of vector space. Here's how they are related:

**Joint Space as a Vector Space:** Joint space is fundamentally a vector space, with each dimension representing a robot's joint or degree of freedom. In this space, the joint values are the vector components. The dimension of the joint space is equal to the number of joints or degrees of freedom in the robot. For example, in a robot with three rotational joints, the joint space is a three-dimensional vector space. Joint values (angles, distances, or other measures) can be thought of as the coordinates of a point within this vector space.

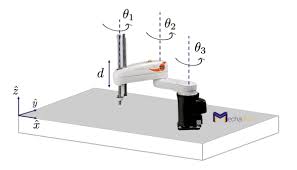
**Vector Space Properties:** Joint space inherits the properties of vector spaces. These properties include closure under addition and scalar multiplication, the existence of a zero vector, and the associativity and commutativity of vector addition. Joint values can be added together or multiplied by scalars, and these operations have mathematical properties similar to those of vectors in a vector space.

**Linear Operations:** In the context of robotics, linear transformations or operations can be applied in joint space. For instance, you can perform linear interpolation between two joint configurations, which corresponds to linear motion in joint space. This is useful for smooth robot motion.

**Robot Control:** In robot control, control algorithms often work with the joint space to determine the necessary joint motions to achieve specific end-effector tasks. The control commands in the joint space are generated to drive the robot's joints to desired positions.

To summarize, the joint space in robotics is theoretically a vector space, with each dimension representing the degree of freedom of a robot joint. This abstraction enables the use of vector space principles and mathematical operations to address a wide range of robotic issues such as motion planning, inverse kinematics, and control.

# **5.5 Task Space**



## **5.5.1 What is Task Space?**

The area where the robot performs its tasks is referred to as the task space in robotics. It is defined only as the dimensions needed to define the task, which are later checked if they match a robot’s operational space or not and if it matches the workspace or not. The task space is a lower-dimensional space than the configuration space, with the number of dimensions equal to the number of degrees of freedom of the robot that are required to perform the task. For example, if the task is to control the position of the tip of a marker on a board, then task space is the Euclidean plane. If the task is to control the position and orientation of a rigid body, then the task space is the 6-dimensional space of rigid body configurations.

## **5.5.2 Relation Between Task Space and Vector Space**

Task space and vector space are related in the context of robotics, particularly when describing the position and orientation of the robot's end-effector. This is how they are related:

**Task Space as a Vector Space:** Task space, which describes the position and orientation of the robot's end-effector in the workspace, can be represented as a vector in a vector space. The Cartesian coordinates (X, Y, Z) of the end-effector's position and orientation angles (e.g., roll, pitch, yaw) can be treated as components of a vector in a three-dimensional vector space. The vector space for task space can be extended to include more dimensions when additional task parameters are considered.

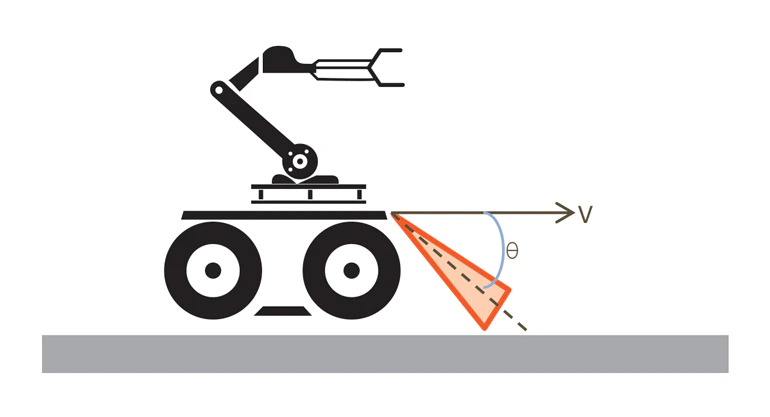
**Vector Space Characteristics:** Task space inherits the properties of vector spaces. These properties include closure under addition and scalar multiplication, the existence of a zero vector, and the associativity and commutativity of vector addition. The vector operations and properties in task space are used to describe the position and orientation of the robot's end-effector in a consistent mathematical framework.

**Linear Operations:** In task space, linear operations can be applied to perform tasks that involve linear interpolation between two end-effector configurations. For instance, this allows for smooth linear motion in task space.

**End-Effector Control:** Task space is a natural choice for defining tasks and control objectives in robotics, as it allows for a more intuitive specification of end-effector behavior. Control commands are often generated in task space to achieve the desired positions and orientations of the end-effector.

All things considered, task space in robotics may be thought of as a specific kind of vector space as it deals with describing the end-effector's location and orientation using Cartesian coordinates and orientation angles, both of which are vector components. For consistent and mathematically well-defined movement and task control of the robot, this vector space framework is necessary.

# **5.6 Velocity Space**

****

## **5.6.1 What is Velocity Space?**

Velocity space, also known as the "operational space," is a concept in robotics that is closely related to the task space and joint space. Velocity space represents the possible velocities of the robot's end-effector in the workspace. In other words, it describes how the end-effector's position and orientation can change over time. The velocity space is a high-dimensional space, with the number of dimensions equal to the number of joints in the robot.

## **5.6.2 Relation Between Velocity Space and Vector Space**

Due to some shared mathematical ideas and qualities, velocity space and vector space in robotics can be coupled in a number of ways:

**Vector Representation:** The velocities of the robot's end-effector are frequently represented as vectors in velocity space. These velocity vectors are usually given in Cartesian coordinates (linear and angular velocities) and can be thought of as vector space elements.

**Vector Space Properties:** Velocity space includes vector space qualities such as closure under addition and scalar multiplication, the presence of a zero vector, and vector addition associativity and commutativity. This allows for the use of vector space operations when working with velocities.

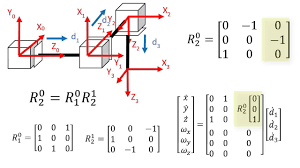
**Linear Operations:** Linear operations can be applied to velocities in velocity space. For example, you can perform linear interpolation between two velocity vectors to create smooth motions or transitions.

**Dimensionality:** The dimensionality of velocity space is determined by the number of degrees of freedom in the robot's end-effector's motion. This dimensionality corresponds to the number of linear and angular velocities that can be controlled. Each dimension in the velocity space represents a specific velocity component.

**Kinematic Constraints:** Constraints on the robot's motion, such as joint limits, can be expressed as constraints in velocity space, similar to how constraints are handled in vector spaces.

In conclusion, velocity space in robotics may be visualized as a vector space where the end-effector velocities of the robot are represented as vectors. The robot's accessibility may be described and controlled using vector space attributes and operations due to this representation. In the context of the robot's end-effector motions, the relationship between velocity space and vector space demonstrates the mathematical basis supporting robot control and motion planning.

# **5.7 Jacobian matrix**

****

## **5.7.1 What is Jacobian Matrix**

In robotics, the Jacobian matrix is a mathematical tool used to relate the joint velocities to the end-effector velocities. It is a matrix of partial derivatives that relates the velocity of the end-effector to the velocity of the joints. The Jacobian matrix is used to calculate the end-effector velocity given the joint velocities.

For a robot that operates in three dimensions, the Jacobian matrix transforms joint velocities into end-effector velocities using the following equation:

​

where represents the end-effector velocities (i.e., linear and angular velocities), represents the joint velocities, and is the Jacobian matrix.

The Jacobian matrix is a useful tool for controlling the velocity of the end-effector of a robotic arm. It is also used for analyzing the performance of a robot and for designing new robots. Also connects the robot's joint space with its task space, allowing for the precise control of the end-effector's motion. It is a central component in the field of robotics and is used in a wide range of applications, including industrial automation, mobile robotics, and manipulator arm control.

## **5.7.2 Relation Between Jacobian matrix and Vector Space**

The Jacobian matrix in robotics is closely related to vector spaces, and this connection is fundamental to understanding how robotic motion and control are described and implemented. Here's the relation between the Jacobian matrix and vector spaces:

Velocity Vectors: The Jacobian matrix is used to relate the joint velocities (typically represented as a vector in joint space) to the end-effector velocities (also represented as a vector in task or velocity space). The end-effector velocity vector describes how the position and orientation of the end-effector change over time.

**Vector Space Representation**: In robotics, both the joint velocities and end-effector velocities are often represented as vectors. Joint velocities describe the change in joint angles or positions, and end-effector velocities describe the change in the position and orientation of the robot's end-effector in Cartesian coordinates. These velocity vectors are elements of their respective vector spaces.

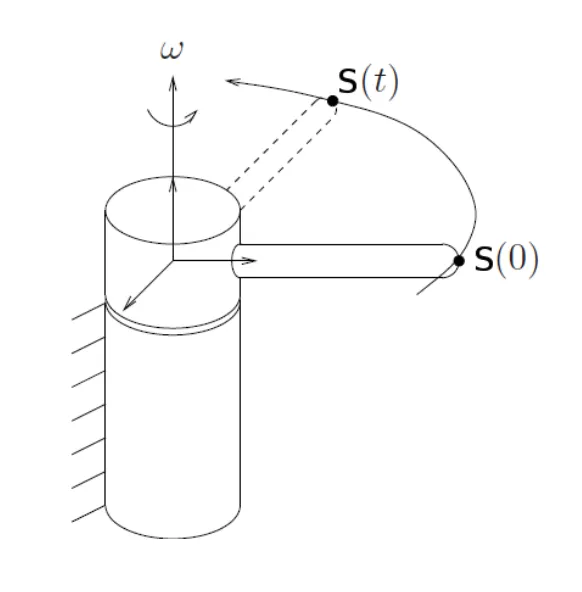
**Linear Transformation:** The Jacobian matrix represents a linear transformation between these two vector spaces. It maps joint velocities (a vector in joint space) to end-effector velocities (a vector in task space). This linear transformation captures how small changes in joint variables correspond to changes in the end-effector's position and orientation.

**Matrix Algebra:** The Jacobian matrix is used to perform matrix-vector multiplication, which allows for the transformation of joint velocities into end-effector velocities. The Jacobian matrix operates on the joint velocity vector to produce the end-effector velocity vector.

**Relationship to Linear Independence:** The Jacobian matrix's columns, when properly chosen, can form a set of linearly independent vectors. This ensures that the Jacobian provides a unique and complete mapping between joint velocities and end-effector velocities.

In summary, the Jacobian matrix serves as a bridge between the joint space (vector space) and the task space (vector space), facilitating the transformation of velocities from one space to another. This mathematical framework is essential for controlling and planning the robot's movements, enabling it to achieve specific tasks and follow desired trajectories in the task space by controlling joint velocities in the joint space.

# **5.8 Lie groups**

****

## **5.8.1 What is Lie Groups**?

Lie groups are mathematical objects that are used to describe the continuous symmetries of geometric objects. In robotics, Lie groups are used to represent the configuration space of a robot. Lie groups are useful in robotics because they provide a way to represent the configuration space of a robot in a way that is amenable to mathematical analysis. They are also used in the design of robot controllers and the development of algorithms for robot motion planning.

## **5.8.2 Relation Between Lie Groups and Vector Space**

Lie groups and vector spaces are closely connected notions in robotics, particularly in the description of robot motion and transformations. The relationship between Lie groups and vector spaces may be described in the following ways:

**Group Actions:** Lie groups are used to represent transformations and configurations in robotics. These transformations often involve actions on vector spaces. For example, a Lie group like the special Euclidean group SE(3) is used to represent rigid-body transformations, which act on 3D vector spaces to translate and rotate objects.

**Tangent Space:** Lie groups have associated tangent spaces, which are vector spaces. The tangent space at the identity element of a Lie group is particularly important. It represents infinitesimal changes in the Lie group's configuration. In robotics, this tangent space often corresponds to the space of velocities (linear and angular velocities) of the robot's end-effector.

**Lie Algebra:** The Lie algebra of a Lie group, which is generated by the tangent space, is itself a vector space. Lie algebras provide a way to analyze the local behavior of Lie groups. In robotics, Lie algebras are used to represent small changes or perturbations in robot configurations, which is crucial for tasks like motion planning and control.

**Lie Group Exponential Map:** The exponential map in Lie theory is used to map elements from the Lie algebra (a vector space) to the Lie group. This mapping allows for the generation of trajectories and motions in the Lie group (e.g., end-effector paths in SE(3)) based on the Lie algebra elements, which can be viewed as velocity vectors.

**Differential Kinematics:** The Lie group structure of rigid-body transformations and the corresponding vector spaces are frequently used to represent the link between the joint velocities (joint space) and the end-effector velocities (task or operational space) in robot kinematics and dynamics.

To summarize, in robotics, Lie groups are utilized to symbolize ongoing changes and arrangements, and vector spaces—specifically, tangent spaces and Lie algebras—are linked to these Lie groups and are employed to depict the local dynamics and modifications of these groups. To effectively describe, analyze, and regulate robot motion in both configuration and velocity spaces, the combination of Lie groups and vector spaces is crucial.

# **5.9 Mathematical Problem and Solution**

Problem and Solution:

Suppose we have a 2-Degrees of Freedom (DOF) planar robot with two joints. We want to find the transformation matrix from the robot's base frame (Frame 0) to the end-effector frame (Frame EE). We have the following Denavit-Hartenberg parameters for each joint:

For Joint 1:  
Theta () =   
d (the distance along the z-axis from the previous joint to the current joint) = 0  
a (the distance along the x-axis from the previous joint to the current joint) = 2  
alpha () =

For Joint 2:  
Theta () =   
d (the distance along the z-axis from the previous joint to the current joint) = 0  
a (the distance along the x-axis from the previous joint to the current joint) = 3  
alpha () =

Now, let's find the transformation matrix from Frame 0 to Frame EE using the Denavit-Hartenberg parameters. The transformation matrix for each joint is given by the following formula:

Now, let's calculate the transformation matrix for Joint 1 (A1) and Joint 2 (A2) using the given DH parameters:

For Joint 1 :

Using the formula, we get:

For Joint 2 ():

Using the formula, we get:

Finally, to find the transformation matrix from Frame 0 to Frame EE (T\_0\_EE), we need to multiply the transformation matrices for each joint in the order from the base to the end-effector:

T\_0\_EE =

So, the transformation matrix T\_0\_EE represents the position and orientation of the end-effector (Frame EE) relative to the base frame (Frame 0) for your 2-DOF planar robot with the given Denavit-Hartenberg parameters.