Vectors

- Arrays of numbers
- Vectors represent a point in a $n$ dimensional space

\[
\begin{pmatrix}
  a_1 \\
  a_2 \\
  \vdots \\
  a_n
\end{pmatrix}
\]
Vectors: Scalar Product

- Scalar-vector product $ka$
- Changes the length of the vector, but **not** its direction
Vectors: Sum

- Sum of vectors (is commutative)

\[
\begin{pmatrix}
    a_1 \\
    a_2 \\
    \vdots \\
    a_n
\end{pmatrix} +
\begin{pmatrix}
    b_1 \\
    b_2 \\
    \vdots \\
    b_n
\end{pmatrix} =
\begin{pmatrix}
    b_1 \\
    b_2 \\
    \vdots \\
    b_n
\end{pmatrix} +
\begin{pmatrix}
    a_1 \\
    a_2 \\
    \vdots \\
    a_n
\end{pmatrix}
\]

- Can be visualized as “chaining” the vectors
Vectors: Dot Product

- Inner product of vectors (yields a scalar)
  \[ a \cdot b = b \cdot a = \sum_i a_i b_i \]

- If \( a \cdot b = 0 \), the two vectors are **orthogonal**
Vectors: Dot Product

- Inner product of vectors (yields a scalar)
  \[ a \cdot b = b \cdot a = \sum a_i b_i \]

- If one of the vectors, e.g., \( a \) has \( \|a\| = 1 \), the product \( a \cdot b \) returns the length of the projection of \( b \) along the direction of \( a \).
A vector $b$ is **linearly dependent** upon $\{a_1, a_2, \ldots, a_n\}$ if

$$b = \sum_i k_i a_i$$
Vectors: Linear Dependence

- A vector $b$ is **linearly dependent** upon $\{a_1, a_2, \ldots, a_n\}$ if

$$b = \sum_{i} k_i a_i$$
Vectors: Linear Independence

- A vector $b$ is **linearly dependent** from $\{a_1, a_2, \ldots, a_n\}$ if

\[
\begin{align*}
\text{if } \quad b &= \sum_i k_i a_i \\
\text{then } b \text{ is } \textbf{independent}\end{align*}
\]

- If there exist no $\{k_i\}$ such that $b = \sum_i k_i a_i$

then $b$ is **independent** from $\{a_i\}$
Matrices

- A matrix is written as a table of values

\[ A = \begin{pmatrix}
  a_{11} & a_{12} & \cdots & a_{1m} \\
  a_{21} & a_{22} & \cdots & a_{2m} \\
  \vdots \\
  a_{n1} & a_{n2} & \cdots & a_{nm}
\end{pmatrix} \quad A : n \times m \]

- 1\textsuperscript{st} index refers to the row
- 2\textsuperscript{nd} index refers to the column
Matrices as Collections of Vectors

- Column vectors

\[
A = \begin{pmatrix}
a_{11} & a_{12} & \cdots & a_{1m} \\
a_{21} & a_{22} & \cdots & a_{2m} \\
\vdots & \vdots & \ddots & \vdots \\
a_{n1} & a_{n2} & \cdots & a_{nm}
\end{pmatrix}
\]
Matrices as Collections of Vectors

- Row vectors

\[ A = \begin{pmatrix}
    a_{11} & a_{12} & \cdots & a_{1m} \\
    a_{21} & a_{22} & \cdots & a_{2m} \\
    \vdots \\
    a_{n1} & a_{n2} & \cdots & a_{nm}
\end{pmatrix} \rightarrow \begin{pmatrix}
    a_1^T \\
    a_2^T \\
    \vdots \\
    a_n^T
\end{pmatrix} \]
Important Matrix Operations

- Multiplication by a scalar
- Sum (commutative, associative)
- Multiplication by a vector
- Product (not commutative)
- Inversion (square, full rank)
- Transposition
 Scalar Multiplication & Sum

- In the **scalar multiplication**, each element of the vector or matrix is multiplied with the scalar
- The **sum of two matrices** is a matrix consisting of the pair-wise sums of the individual entries
Matrix Vector Product

- The $i^{th}$ component of $A \cdot b$ is the dot product $a^T_{i*} \cdot b$

- The vector $A \cdot b$ is linearly dependent from $\{a_{*i}\}$ with coefficients $\{b_i\}$

$$A \cdot b = \left( \begin{array}{c} a^T_{1*} \\ a^T_{2*} \\ \vdots \\ a^T_{n*} \end{array} \right) \cdot b = \left( \begin{array}{c} a^T_{1*} \cdot b \\ a^T_{2*} \cdot b \\ \vdots \\ a^T_{n*} \cdot b \end{array} \right) = \sum_k a_{*k} \cdot b_k$$

row vectors   column vectors
Matrix Vector Product

- If the column vectors of $A$ represent a reference system, the product $A \cdot b$ computes the **global transformation** of the vector $b$ according to $\{a_{*i}\}$.
Matrix Matrix Product

Can be defined through

- the dot product of row and column vectors
- the linear combination of the columns of $A$ scaled by the coefficients of the columns of $B$

\[
C = AB
\]

\[
= \begin{pmatrix}
    a_{1*}^T \cdot b_{*1} & a_{1*}^T \cdot b_{*2} & \cdots & a_{1*}^T \cdot b_{*m} \\
    a_{2*}^T \cdot b_{*1} & a_{2*}^T \cdot b_{*2} & \cdots & a_{2*}^T \cdot b_{*m} \\
    \vdots & \vdots & \ddots & \vdots \\
    a_{n*}^T \cdot b_{*1} & a_{n*}^T \cdot b_{*2} & \cdots & a_{n*}^T \cdot b_{*m}
\end{pmatrix}
\]

\[
= \begin{pmatrix}
    A \cdot b_{*1} & A \cdot b_{*2} & \cdots & A \cdot b_{*m}
\end{pmatrix}
\]
Matrix Matrix Product

- If we consider the second interpretation, we see that the columns of $C$ are the “global transformations” of the columns of $B$ through $A$

- All the interpretations made for the matrix vector product hold

\[
C = AB \\
= \left( \begin{array}{ccc}
A \cdot b_{*1} & A \cdot b_{*2} & \ldots & A \cdot b_{*m} \\
\end{array} \right) \\
c_{*i} = A \cdot b_{*i}
\]
Inverse

\[
AB = I
\]

- If \( A \) is a square matrix of full rank, then there is a unique matrix \( B = A^{-1} \) such that \( AB = I \) holds.

- The \( i^{th} \) row of \( A \) and the \( j^{th} \) column of \( A^{-1} \) are
  - orthogonal (if \( i \neq j \))
  - or their dot product is 1 (if \( i = j \))
Matrix Inversion

\[ AA^{-1} = I \]

- The \( i^{th} \) column of \( A^{-1} \) can be found by solving the following linear system:

\[ A_{\*i}^{-1} = i_{*i} \]

This is the \( i^{th} \) column of the identity matrix.
Linear Systems (1)

\[ Ax = b \]

- A set of linear equations
- Solvable by Gaussian elimination
- Many efficient solvers exist, e.g., conjugate gradients, sparse Cholesky decomposition
Linear Systems (2)

\[ Ax = b \]

Notes:

- One can obtain a reduced system \((A', b')\) by considering the matrix \((A, b)\) and suppressing all the rows which are linearly dependent.

- Let \(A'x = b'\) be the reduced system with \(A' : n' \times m\) and \(b' : n' \times 1\) and \(\text{rank}(A') = \min(n', m)\).

- The system might be either overdetermined \((n' > m)\) or underconstrained \((n' < m)\)
Overdetermined Systems

- More independent equations than variables
- An overdetermined system does not admit an exact solution
- "Least-squares" problem
Underconstrained Systems

- More variables than (independent) equations
- The number of linearly independent rows (or columns) of $A'$ is smaller than the dimension of $b'$
- An underconstrained system admits infinite solutions
- The degree of these infinite solutions is $\text{cols}(A') - \text{rows}(A')$
Orthonormal Matrix

- A matrix $Q$ is **orthonormal** iff its column (row) vectors represent an **orthonormal basis**

$$q_{*i}^T \cdot q_{*j} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}, \forall i, j$$

- The transpose is the inverse

$$QQ^T = Q^T Q = I$$
Rotation Matrix (Orthonormal)

- **2D Rotations:**
  \[ R(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \]

- **3D Rotations along the main axes**
  \[
  R_x(\theta) = \begin{bmatrix}
  1 & 0 & 0 \\
  0 & \cos(\theta) & -\sin(\theta) \\
  0 & \sin(\theta) & \cos(\theta)
  \end{bmatrix} \quad R_y(\theta) = \begin{bmatrix}
  \cos(\theta) & 0 & \sin(\theta) \\
  0 & 1 & 0 \\
  -\sin(\theta) & 0 & \cos(\theta)
  \end{bmatrix} \\
  R_z(\theta) = \begin{bmatrix}
  \cos(\theta) & -\sin(\theta) & 0 \\
  \sin(\theta) & \cos(\theta) & 0 \\
  0 & 0 & 1
  \end{bmatrix}
  \]

- The inverse is the transpose (efficient)

- **IMPORTANT:** Rotations are not commutative!

  \[
  R_x(\frac{\pi}{4}) \cdot R_y(\frac{\pi}{4}) \cdot \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \neq R_y(\frac{\pi}{4}) \cdot R_x(\frac{\pi}{4}) \cdot \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}
  \]
Jacobian Matrix

- It is a **non-square matrix** $m \times n$ in general
- Given a vector-valued function

\[
    f(x) = \begin{bmatrix}
        f_1(x) \\
        f_2(x) \\
        \vdots \\
        f_m(x)
    \end{bmatrix}
\]

- Then, the **Jacobian matrix** is defined as

\[
    F_x = \begin{bmatrix}
        \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\
        \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\
        \vdots & \vdots & \cdots & \vdots \\
        \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n}
    \end{bmatrix}
\]
Jacobian Matrix

- Orientation of the tangent plane to the vector-valued function at a given point

- Generalizes the gradient of a scalar valued function
Quadratic Forms

- Many functions can be locally approximated with a **quadratic form**

\[
    f(x) = \sum_{i,j} a_{ij} x_i x_j + \sum_i b_i x_i + c
\]

\[
    = x^T A x + bx + c
\]

- Often, one is interested in finding the minimum (or maximum) of a quadratic form, i.e.,

\[
    \hat{x} = \arg\min_x f(x)
\]
Quadratic Forms

- Question: How to efficiently compute a solution to this minimization problem

\[ \hat{x} = \arg\min_x f(x) \]

- At the minimum, we have \( f'(\hat{x}) = 0 \)
- By using the definition of matrix product, we can compute \( f' \)

\[
\begin{align*}
  f(x) &= x^T A x + b x + c \\
  f'(x) &= A^T x + A x + b
\end{align*}
\]
Quadratic Forms

▪ The minimum of $f(x) = x^T A x + b x + c$ is where its derivative is 0

$$0 = A^T x + Ax + b$$

▪ Thus, we can solve the system

$$(A^T + A)x = -b$$

▪ If the matrix is symmetric, the system becomes

$$2Ax = -b$$

▪ Solving that, leads to the minimum
Conclusions

These LA basics

- help to understand the next chapters
- are needed to solve the exercises