Vectors

- Arrays of numbers
- They 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 $k \cdot \mathbf{a}$
- 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 (is a scalar)

\[ a \cdot b = b \cdot a = \sum_{i} a_i \cdot b_i \]

- If one of the two vectors has \(|a| = 1\) the inner product \(a \cdot b\) returns the length of the projection of \(b\) along the direction of \(a\)

- If \(a \cdot b = 0\) the two vectors are orthogonal
Vectors: Linear (In)Dependence

- A vector \( \mathbf{b} \) is **linearly dependent** from \( \{ \mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n \} \): 
  \[ \mathbf{b} = \sum k_i \cdot \mathbf{a}_i \]

- In other words if \( \mathbf{b} \) can be obtained by summing up the \( \mathbf{a}_i \) properly scaled.

- If do not exist \( \{ k_i \} \) such that \( \mathbf{b} = \sum k_i \cdot \mathbf{a}_i \) then \( \mathbf{b} \) is independent from \( \{ \mathbf{a}_i \} \)
Vectors: Linear (In)Dependence

- A vector $b$ is **linearly dependent** from \( \{a_1, a_2, \ldots, a_n\} \): 
  \[ b = \sum_i k_i \cdot a_i \]

- In other words if $b$ can be obtained by summing up the $a_i$ properly scaled.

- If do not exist \( \{k_i\} \) such that 
  \[ b = \sum_i k_i \cdot a_i \]
  then $b$ is independent from \( \{a_i\} \)
Matrices

- A matrix is written as a table of values
- Can be used in many ways:

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

- 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

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

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

- The $i$ 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 $\{b_i\}$

\[
A \cdot b = \begin{pmatrix}
a^T_{1*} \\
a^T_{2*} \\
\vdots \\
a^T_{n*}
\end{pmatrix} \cdot b = \begin{pmatrix}
a^T_{1*} \cdot b \\
a^T_{2*} \cdot b \\
\vdots \\
a^T_{n*} \cdot b
\end{pmatrix} = \sum_k a_{*k} \cdot b_k
\]
Matrix Vector Product

- If the column vectors represent a reference system, the product $A \cdot b$ computes the global transformation of the vector $\mathbf{b}$ according to $\{a_i\}$. 

![Diagram](image.png)
Matrix Vector Product

- Each $a_{i,j}$ can be seen as a linear mixing coefficient that tells how contributes to $(A \cdot b)_j$

- Example: Jacobian of a multi-dimensional function

\[
y = f(x) = \begin{pmatrix}
f_1(x) \\
f_2(x) \\
\vdots \\
f_n(x)
\end{pmatrix} \quad J_f = \begin{pmatrix}
\frac{df_1}{dx_1} & \frac{df_1}{dx_2} & \cdots & \frac{df_1}{dx_m} \\
\frac{df_2}{dx_1} & \frac{df_2}{dx_2} & \cdots & \frac{df_2}{dx_m} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{df_n}{dx_1} & \frac{df_n}{dx_2} & \cdots & \frac{df_n}{dx_m}
\end{pmatrix}
\]
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 $B$.

$$
C = A \cdot B
= \begin{pmatrix}
    a_{1*}^T \cdot b_{*1} & a_{1*}^T \cdot b_{*2} & \cdots & a_{1*}^T \cdot b_{*m} \\
    a_{2*}^T \cdot b_{*1} & a_{2*}^T \cdot b_{*2} & \cdots & a_{2*}^T \cdot b_{*m} \\
    \cdots & & & \\
    a_{n*}^T \cdot b_{*1} & a_{n*}^T \cdot b_{*2} & \cdots & a_{n*}^T \cdot b_{*m}
\end{pmatrix}
= \begin{pmatrix}
    A \cdot b_{*1} & A \cdot b_{*2} & \cdots & A \cdot b_{*m}
\end{pmatrix}
$$
Matrix Matrix Product

- If we consider the second interpretation we see that the columns of $C$ are the projections of the columns of $B$ through $A$.
- All the interpretations made for the matrix vector product hold.

$$C = A \cdot B$$

$$= \left( \begin{array}{c} A \cdot b_{*1} \\ A \cdot b_{*2} \\ \vdots \\ A \cdot b_{*m} \end{array} \right)$$

$$c_{*i} = A \cdot b_{*i}$$
Linear Systems

\[ Ax = b \]

- Interpretations:
  - Find the coordinates \( x \) in the reference system of \( A \) such that \( b \) is the result of the transformation of \( Ax \).
  - Many efficient solvers
    - Conjugate gradients
    - Sparse Cholesky Decomposition (if SPD)
    - …
  - The system may be **over** or **under** constrained.
  - One can obtain a reduced system \((A' b')\) by considering the matrix \((A b)\) and suppressing all the rows which are linearly dependent.
Linear Systems

- The system is **over-constrained** if the number of linearly independent columns (or rows) of $A'$ is greater than the dimension of $b'$.
- An **over-constrained** system does not admit a solution, however one may find a minimum norm solution by pseudo inversion

$$x = \arg\min_{x} ||A'x - b'|| = (A'^TA')^{-1}A'^Tb'$$
The system is under-constrained if the number of linearly independent columns (or rows) of $A'$ is smaller than the dimension of $b'$.

An under-constrained admits infinite solutions. The degree of infinity is $\text{rank}(A') - \text{dim}(b')$.

The rank of a matrix is the maximum number of linearly independent rows or columns.
Matrix Inversion

\[ \mathbf{A} \cdot \mathbf{B} = \mathbf{I} \]

- If \( \mathbf{A} \) is a square matrix of full rank, then there is a unique matrix \( \mathbf{B} = \mathbf{A}^{-1} \) such that the above equation holds.
- The \( i^{th} \) row of \( \mathbf{A} \) is and the \( j^{th} \) column of \( \mathbf{A}^{-1} \) are:
  - orthogonal, if \( i = j \)
  - their scalar product is 1, otherwise.
- The \( i^{th} \) column of \( \mathbf{A}^{-1} \) can be found by solving the following system:

\[
\mathbf{A} \cdot \mathbf{a}^{-1} \hat{i}_i = \mathbf{i} \hat{i}_i \quad \text{This is the} \ i^{th} \ \text{column of the identity matrix}
\]
Trace

- Only defined for **square matrices**
- **Sum** of the elements on the main diagonal, that is
  \[
  \text{tr}(A) = a_{11} + a_{22} + \cdots + a_{nn} = \sum_{i=1}^{n} a_{ii}
  \]
- It is a linear operator with the following properties
  - Additivity: \( \text{tr}(A + B) = \text{tr}(A) + \text{tr}(B) \)
  - Homogeneity: \( \text{tr}(c \cdot A) = c \cdot \text{tr}(A) \)
  - Pairwise commutative: \( \text{tr}(AB) = \text{tr}(BA), \quad \text{tr}(ABC) \neq \text{tr}(ACB) \)
- Trace is similarity invariant \( \text{tr}(P^{-1}AP) = \text{tr}((AP^{-1})P) = \text{tr}(A) \)
- Trace is transpose invariant \( \text{tr}(A) = \text{tr}(A^T) \)
Rank

- **Maximum** number of linearly independent rows (columns)
- Dimension of the **image** of the transformation \( f(x) = Ax \)

- When \( A \) is \( m \times n \) we have
  - \( \text{rank}(A) \geq 0 \) and the equality holds iff \( A \); the null matrix
  - \( \text{rank}(A) \leq \min(m, n) \)
  - \( f(x) \): **injective** iff \( \text{rank}(A) = n \)
  - \( f(x) \): **surjective** iff \( \text{rank}(A) = m \)
  - if \( m = n \) \( f(x) \); **bijective** and \( A \)** invertible iff \( \text{rank}(A) = n \)

- Computation of the rank is done by
  - Perform Gaussian elimination on the matrix
  - Count the number of non-zero rows
Determinant

- Only defined for **square matrices**
- Remember? \( A \cdot A^{-1} = I \) if and only if \( \det(A) \neq 0 \)
- For \( 2 \times 2 \) matrices:
  Let \( A = [a_{ij}] \) and \( |A| = \det(A) \):
  \[
  \begin{vmatrix}
    a_{11} & a_{12} \\
    a_{21} & a_{22}
  \end{vmatrix}
  = a_{11} \cdot a_{22} - a_{12} \cdot a_{21}
  \]

- For \( 3 \times 3 \) matrices:
  \[
  \begin{vmatrix}
    a_{11} & a_{12} & a_{13} \\
    a_{21} & a_{22} & a_{23} \\
    a_{31} & a_{32} & a_{33}
  \end{vmatrix}
  = a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{21}a_{32}
  - a_{11}a_{23}a_{32} - a_{12}a_{21}a_{33} - a_{13}a_{22}a_{11}
  \]
**Determinant**

- For **general** $n \times n$ matrices?

Let $A_{ij}$ be the submatrix obtained from $A$ by deleting the $i$-th row and the $j$-th column.

\[
\begin{bmatrix}
1 & 2 & 5 & 0 \\
2 & 3 & 4 & -1 \\
-5 & 8 & 0 & 0 \\
0 & 4 & -2 & 0
\end{bmatrix}
\rightarrow
\begin{bmatrix}
1 & 5 & 0 \\
2 & 4 & -1 \\
0 & -2 & 0
\end{bmatrix}
\]

Rewrite determinant for $3 \times 3$ matrices:

\[
det(A_{3\times3}) = a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{21}a_{32} - a_{11}a_{23}a_{32} - a_{12}a_{21}a_{33} - a_{13}a_{22}a_{11} = a_{11} \cdot det(A_{11}) - a_{12} \cdot det(A_{12}) + a_{13} \cdot det(A_{13})
\]
Determinant

- For general $n \times n$ matrices?

$$det(A) = a_{11} det(A_{11}) - a_{12} det(A_{12}) + \ldots + (-1)^{1+n} a_{1n} det(A_{1n})$$

$$= \sum_{j=1}^{n} (-1)^{1+j} a_{1j} det(A_{1j})$$

Let $C_{ij} = (-1)^{i+j} det(A_{ij})$ be the $(i,j)$-cofactor, then

$$det(A) = a_{11} C_{11} + a_{12} C_{12} + \ldots + a_{1n} C_{1n}$$

$$= \sum_{j=1}^{n} a_{1j} C_{1j}$$

This is called the cofactor expansion across the first row.
Problem: Take a 25 x 25 matrix (which is considered small). The cofactor expansion method requires n! multiplications. For n = 25, this is $1.5 \times 10^{25}$ multiplications for which a today supercomputer would take 500,000 years.

There are much faster methods, namely using Gauss elimination to bring the matrix into triangular form.

Then:

$$A = \begin{bmatrix} d_1 & * & * & * \\ 0 & d_2 & * & * \\ 0 & 0 & d_3 & * \\ 0 & 0 & 0 & d_4 \end{bmatrix} \quad \text{det}(A) = \prod_{i=1}^{n} d_i$$

Because for triangular matrices (with $A$ being invertible), the determinant is the product of diagonal elements.
Determinant: Properties

- **Row operations** \( (A \text{ still a } n \times n \text{ square matrix}) \)
  - If \( B \) results from \( A \) by interchanging two rows, then \( \det(B) = -\det(A) \)
  - If \( B \) results from \( A \) by multiplying one row with a number \( c \), then \( \det(B) = c \cdot \det(A) \)
  - If \( B \) results from \( A \) by adding a multiple of one row to another row, then \( \det(B) = \det(A) \)

- **Transpose:** \( \det(A^T) = \det(A) \)

- **Multiplication:** \( \det(A \cdot B) = \det(A) \cdot \det(B) \)

- Does **not** apply to addition! \( \det(A + B) \neq \det(A) + \det(B) \)
Determinant: Applications

- **Find the inverse** $A^{-1}$ using Cramer’s rule
  $A^{-1} = \frac{\text{adj}(A)}{\text{det}(A)}$
  with $\text{adj}(A)$ being the adjugate of $A$

- **Compute Eigenvalues**
  Solve the characteristic polynomial $\text{det}(A - \lambda \cdot I) = 0$

- **Area and Volume:** $\text{area} = |\text{det}(A)|$

\[
A = \begin{bmatrix}
a & b \\
c & d \\
\end{bmatrix}
\]

\[
A = \begin{bmatrix}
a & b & c \\
d & e & f \\
g & h & i \\
\end{bmatrix}
\]

($r_i$ is $i$-th row)
Orthogonal matrix

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

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

- As linear transformation, it is **norm** preserving, and acts as an isometry in Euclidean space (rotation, reflection)

- Some properties:
  - The transpose is the inverse $QQ^T = Q^TQ = I$
  - Determinant has unity norm (§ 1)

\[
1 = \det(I) = \det(Q^TQ) = \det(Q)\det(Q^T) = (\det(Q))^2
\]
Rotational matrix

- **Important** in robotics
  - 2D Rotations \( R(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \)
  - 3D Rotations along the main axes
    \[
    R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix} \\
    R_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix}
    \]
- **IMPORTANT**: Rotations are **not commutative**

\[
R_x\left(\frac{\pi}{4}\right) \cdot R_y\left(\frac{\pi}{4}\right) = \begin{bmatrix} 0.707 & 0 & -0.707 \\ -0.5 & 0.707 & -0.5 \\ 0.5 & 0.707 & 0.5 \end{bmatrix}, \quad R_x\left(\frac{\pi}{4}\right) \cdot R_y\left(\frac{\pi}{4}\right) \cdot \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} -1.414 \\ 0.586 \\ 3.414 \end{bmatrix}
\]

\[
R_y\left(\frac{\pi}{4}\right) \cdot R_x\left(\frac{\pi}{4}\right) = \begin{bmatrix} 0.707 & -0.5 & -0.5 \\ 0 & 0.707 & -0.707 \\ 0.707 & 0.5 & 0.5 \end{bmatrix}, \quad R_y\left(\frac{\pi}{4}\right) \cdot R_x\left(\frac{\pi}{4}\right) \cdot \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} -1.793 \\ 0.707 \\ 3.207 \end{bmatrix}
\]
A general and easy way to describe a 3D transformation is via matrices.

\[
A = \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix} \quad A^{-1} = \begin{pmatrix} R^T & -R^T t \\ 0 & 1 \end{pmatrix} \quad p = \begin{pmatrix} t \\ 1 \end{pmatrix}
\]

- Homogeneous behavior in 2D and 3D
- Takes naturally into account the non-commutativity of the transformations
Combining Transformations

- A simple interpretation: chaining of transformations (represented as homogeneous matrices)
  - Matrix $A$ represents the pose of a robot in the space
  - Matrix $B$ represents the position of a sensor on the robot
  - The sensor perceives an object at a given location $p$, in its own frame [the sensor has no clue on where it is in the world]
  - Where is the object in the global frame?
Combining Transformations

- A simple interpretation: chaining of transformations (represented as homogeneous matrices)
  - Matrix $A$ represents the pose of a **robot** in the space
  - Matrix $B$ represents the position of a sensor on the robot
  - The **sensor** perceives an **object** at a given location $p$, in its own frame [the sensor has no clue on where it is in the world]
  - Where is the object in the global frame?

$Bp$ gives me the pose of the object wrt the robot
Combining Transformations

- A simple interpretation: chaining of transformations (represented as homogeneous matrices)
  - Matrix $A$ represents the pose of a **robot** in the space
  - Matrix $B$ represents the position of a sensor on the robot
  - The **sensor** perceives an **object** at a given location $p$, in its own frame [the sensor has no clue on where it is in the world]
  - Where is the object in the global frame?

$Bp$ gives me the pose of the object wrt the robot

$ABp$ gives me the pose of the object wrt the world
Symmetric matrix

- A matrix is symmetric if \( A = A^T \) e.g.
  \[
  \begin{bmatrix}
  1 & 4 & -2 \\
  4 & -1 & 3 \\
  -2 & 3 & 5 \\
  \end{bmatrix}
  \]

- A matrix is anti-symmetric if \( A = -A^T \) e.g.
  \[
  \begin{bmatrix}
  0 & 4 & -2 \\
  -4 & 0 & 3 \\
  2 & -3 & 0 \\
  \end{bmatrix}
  \]

- Every symmetric matrix:
  - can be diagonalizable \( D = QAQ^T \) where \( D \); a diagonal matrix of eigenvalues and \( Q \); an orthogonal matrix whose columns are the eigenvectors of \( A \)
  - define a quadratic form \( q(x) = x^T Ax = \sum_{i,j=1}^{n} a_{ij}x_ix_j \)
Positive definite matrix

- The analogous of positive number

- Definition
  \[ M > 0 \text{ iff } z^T M z > 0 \forall z > 0 \]

- Examples
  \[ M_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = z_1^2 + z_2^2 > 0 \]
  \[ M_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = 2z_1z_2 < 0, \ z_1 = -z_2 \]
Positive definite matrix

- **Properties**
  - **Invertible**, with positive definite inverse
  - All **eigenvalues** > 0
  - **Trace** is > 0
  - For any spd \( A \) \( AAT, A^TA \) are positive definite
  - **Cholesky decomposition** \( A = LL^T \)
  - **Partial ordering**: \( M > N \) iff \( M - N > 0 \)
  - If \( M > N > 0 \) we have \( N^{-1} > M - 1 > 0 \)
  - If \( M, N > 0 \) then
    - \( M + N > 0 \)
    - \( MNM, NNM > 0 \)
Jacobian Matrix

• It’s a **non-square matrix** $n \times m$ in general

• Suppose you have a vector-valued function $f(x) = \begin{bmatrix} f_1(x) \\ f_2(x) \end{bmatrix}$

• Let the **gradient operator** be the vector of (first-order) partial derivatives

\[
\nabla_x = \begin{bmatrix} \frac{\partial}{\partial x_1} & \frac{\partial}{\partial x_2} & \cdots & \frac{\partial}{\partial x_n} \end{bmatrix}^T
\]

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

\[
F_x = \begin{bmatrix} f_1(x) \\ f_2(x) \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial}{\partial x_1} & \cdots & \frac{\partial}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \cdots & \frac{\partial f_2}{\partial x_n} \end{bmatrix}
\]
Jacobian Matrix

• It’s the orientation of the **tangent plane** to the vector-valued function at a given point

• Generalizes the **gradient** of a scalar valued function

• Heavily used for **first-order error propagation**

\[
C_{out} = F \cdot C_{in} \cdot F^T
\]

→ See later in the course
Quadratic Forms

- Many important functions can be locally approximated with a quadratic form.

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

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

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

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

- How can we use the matrix properties to quickly 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 \( f' \)

\[ f(x) = x^T Ax + bx + c \]
\[ f'(x) = A^T x + Ax + b \]
The minimum of \( f(x) = x^TAx + bx + c \) is where its derivative is set to 0

\[
0 = A^T x + Ax + b
\]

Thus we can solve the system

\[
(A^T + A)^T x = -b
\]

If the matrix is symmetric, the system becomes

\[
2Ax = -b
\]