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 (is a scalar)
  \[ a \cdot b = b \cdot a = \sum_{i} a_i b_i \]

- If one of the two vectors, e.g. \( \mathbf{a} \), has \( \|\mathbf{a}\| = 1 \) then the inner product \( \mathbf{a} \cdot \mathbf{b} \) returns the length of the projection of \( \mathbf{b} \) along the direction of \( \mathbf{a} \).

- If \( \mathbf{a} \cdot \mathbf{b} = 0 \), the two vectors are **orthogonal**.

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>( \mathbf{a} \cdot \mathbf{b} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( |\mathbf{a}| = 1 )</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Vectors: Linear (In)Dependence

- A vector $\mathbf{b}$ is **linearly dependent** from $\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n\}$ if $\mathbf{b} = \sum_i k_i \mathbf{a}_i$

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

- If there exist no $\{k_i\}$ such that $\mathbf{b} = \sum_i k_i \mathbf{a}_i$ then $\mathbf{b}$ is independent from $\{\mathbf{a}_i\}$
A vector $\mathbf{b}$ is **linearly dependent** from $\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n\}$ if

$$\mathbf{b} = \sum_i k_i \mathbf{a}_i$$

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

If there exist no $\{k_i\}$ such that

$$\mathbf{b} = \sum_i k_i \mathbf{a}_i$$

then $\mathbf{b}$ is independent from $\{\mathbf{a}_i\}$.
Matrices

- A matrix is written as a table of values

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

- 1st index refers to the row
- 2nd index refers to the column
- Note: a d-dimensional vector is equivalent to a dx1 matrix
Matrices as Collections of Vectors

- Column vectors
Matrices as Collections of Vectors

- Row vectors

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

\[ \rightarrow \begin{pmatrix}
\begin{array}{c}
a_1^T \\
a_2^T \\
\vdots \\
a_n^T
\end{array}
\end{pmatrix} \]
Important Matrices 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, every element of the vector or matrix is multiplied with the scalar.

- The sum of two vectors is a vector consisting of the pair-wise sums of the individual entries.

- 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 $Ab$ is the dot product $a_{i*}^T \cdot b$.
- The vector $Ab$ is linearly dependent from the column vectors $\{a_{*i}\}$ with coefficients $\{b_i\}$.

\[
Ab = \left( \begin{array}{c} a_{1*}^T \\ a_{2*}^T \\ \vdots \\ a_{n*}^T \end{array} \right) \cdot b = \left( \begin{array}{c} a_{1*}^T \cdot b \\ a_{2*}^T \cdot b \\ \vdots \\ a_{n*}^T \cdot b \end{array} \right) = \sum_k a_{*k} b_k
\]
Matrix Vector Product

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

\[
C = AB = \begin{pmatrix}
    a^T_{1*} \cdot b_{*1} & a^T_{1*} \cdot b_{*2} & \cdots & a^T_{1*} \cdot b_{*m} \\
    a^T_{2*} \cdot b_{*1} & a^T_{2*} \cdot b_{*2} & \cdots & a^T_{2*} \cdot b_{*m} \\
    \vdots & \vdots & \ddots & \vdots \\
    a^T_{n*} \cdot b_{*1} & a^T_{n*} \cdot b_{*2} & \cdots & a^T_{n*} \cdot b_{*m}
\end{pmatrix} = \begin{pmatrix}
    Ab_{*1} & Ab_{*2} & \cdots & Ab_{*m}
\end{pmatrix}
\]
Matrix Matrix Product

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

\[
C = AB \\
= \begin{pmatrix} Ab_1 & Ab_2 & \ldots & Ab_m \end{pmatrix} \\
c_{*i} = Ab_{*i}
\]
**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 \( A \) is the null matrix
  - \( \text{rank}(A) \leq \min(m, n) \)

- Computation of the rank is done by
  - Gaussian elimination on the matrix
  - Counting the number of non-zero rows
**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 $A$ is 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

\[ AB = I \]

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

\[ A a_{1}^{-1} i = i \]

This is the \( i^{th} \) column of the identity matrix.
Determinant (det)

- Only defined for **square matrices**
- The inverse of $A$ exists if and only if $det(A) \neq 0$
- For $2 \times 2$ matrices:
  Let $A = [a_{ij}]$ and $|A| = det(A)$, then
  
  $$
  \begin{vmatrix}
  a_{11} & a_{12} \\
  a_{21} & a_{22}
  \end{vmatrix}
  = a_{11} \cdot a_{22} - a_{12} \cdot a_{21}
  $$

- For $3 \times 3$ matrices the Sarrus rule holds:

  $$
  \begin{vmatrix}
  a_{11} & a_{12} & a_{13} \\
  a_{21} & a_{22} & a_{23} \\
  a_{31} & a_{32} & a_{33}
  \end{vmatrix}
  = a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{21}a_{32}
  - a_{11}a_{23}a_{32} - a_{12}a_{21}a_{33} - a_{13}a_{22}a_{11}
  $$
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?

$$
\begin{align*}
\det(A) &= a_{11} \det(A_{11}) - a_{12} \det(A_{12}) + \ldots + (-1)^{1+n} a_{1n} \det(A_{1n}) \\
&= \sum_{j=1}^{n} (-1)^{1+j} a_{1j} \det(A_{1j})
\end{align*}
$$

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

$$
\begin{align*}
\det(A) &= a_{11} C_{11} + a_{12} C_{12} + \ldots + a_{1n} C_{1n} \\
&= \sum_{j=1}^{n} a_{1j} C_{1j}
\end{align*}
$$

This is called the **cofactor expansion** across the first row.
**Determinant**

- **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.

\[
A = \begin{bmatrix}
  d_1 & * & * & * \\
  0 & d_2 & * & * \\
  0 & 0 & d_3 & * \\
  0 & 0 & 0 & d_4
\end{bmatrix}
\]

\[
det(A) = \prod_{i=1}^{n} d_i
\]

Because for **triangular matrices** the determinant is the product of diagonal elements.
Determinant: Properties

- **Row operations** ($A$ is still a $n \times n$ 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

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

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

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

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

($r_i$ is $i$-th row)
Orthonormal Matrix

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

\[
q^T_i \cdot q_j = \begin{cases} 
1 & \text{if } i = j \\
0 & \text{if } i \neq j 
\end{cases}, \forall i, j
\]

- As linear transformation, it is **norm** preserving

- Some properties:
  - The transpose is the inverse $QQ^T = Q^TQ = I$
  - Determinant has unity norm ($\S$ 1)
    \[
    1 = \det(I) = \det(Q^TQ) = \det(Q)\det(Q^T) = \det(Q)^2
    \]
**Rotation Matrix**

- A Rotation matrix is an orthonormal matrix with $\det = +1$
  - 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

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

$$
R_y\left(\frac{\pi}{4}\right) \cdot R_x\left(\frac{\pi}{4}\right) = \begin{bmatrix}
0.707 & -0.5 & -0.5 \\
0 & 0.707 & -0.707 \\
0.707 & 0.5 & 0.5
\end{bmatrix},
R_y\left(\frac{\pi}{4}\right) \cdot R_x\left(\frac{\pi}{4}\right) \cdot \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} -1.793 \\ 0.707 \\ 3.207 \end{bmatrix}
$$
Matrices to Represent Affine Transformations

- A general and easy way to describe a 3D transformation is via matrices.

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

- Takes naturally into account the non-commutativity of the transformations.
- Homogeneous coordinates.
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 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 the pose of the object wrt the robot

$ABp$ gives the pose of the object wrt the world
Positive Definite Matrix

- The analogous of positive number

- Definition \( M > 0 \) iff \( z^T M z > 0 \forall z \neq 0 \)

- Example

\[
M_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = z_1^2 + z_2^2 > 0
\]
**Positive Definite Matrix**

- **Properties**
  - *Invertible*, with positive definite inverse
  - All real **eigenvalues** $> 0$
  - **Trace** is $> 0$
  - **Cholesky** decomposition $A = LL^T$
Linear Systems (1)

\[ Ax = b \]

Interpretations:
- A set of linear equations
- A way to find the coordinates \( x \) in the reference system of \( A \) such that \( b \) is the result of the transformation of \( Ax \)
- Solvable by Gaussian elimination
Linear Systems (2)

$Ax = b$

Notes:

- Many efficient solvers exist, e.g., conjugate gradients, sparse Cholesky decomposition
- One can obtain a reduced system $(A', b')$ by considering the matrix $(A, b)$ and suppressing all the rows which are linearly dependent
- Let $A'x = b'$ the reduced system with $A':n'x m$ and $b':n'x 1$ and rank $A' = \min(n', m)$
- The system might be either over-constrained ($n' > m$) or under-constrained ($n' < m$)
Over-Constrained Systems

- “More (indep) equations than variables”
- An over-constrained system does not admit an **exact solution**
- However, if \( \text{rank } A' = \text{cols}(A) \) one often computes a **minimum norm solution**

\[
x = \arg\min_x \|A'x - b'\|
\]

Note: rank = Maximum number of linearly independent rows/columns
Under-Constrained Systems

- “More variables than (indep) equations”
- The system is **under-constrained** if the number of linearly independent rows of $A'$ is smaller than the dimension of $b'$
- An under-constrained system admits infinite solutions
- The degree of these infinite solutions is $\text{cols}(A') - \text{rows}(A')$
Jacobian Matrix

- It is a **non-square matrix** $n \times m$ 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

  $$\mathbf{F}_x = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$
Jacobian Matrix

- It is the orientation of the **tangent plane** to the vector-valued function at a given point.

- **Generalizes the gradient** of a scalar valued function.
Further Reading

- A “quick and dirty” guide to matrices is the Matrix Cookbook available at:
  http://matrixcookbook.com