Sapienza Università di Roma
Dipartimento di Informatica e Sistemistica Antonio Ruberti
Uni Freiburg
AIS Autonomous Intelligent Systems

Least Squares and SLAM
A Compact Course on Linear Algebra

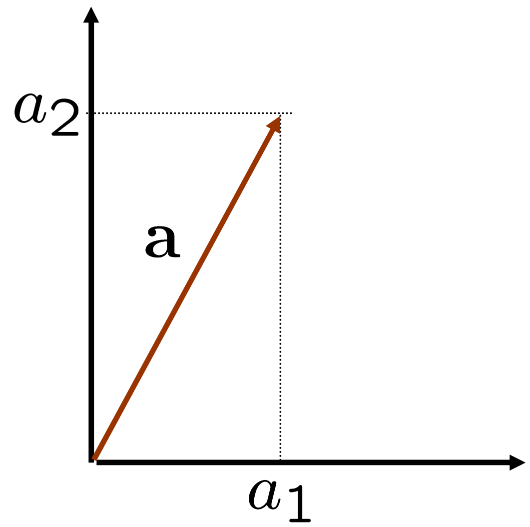
Giorgio Grisetti

Part of the material of this course is taken from the Robotics 2 lectures given by G.Grisetti, W.Burgard, C.Stachniss, K.Arras, D. Tipaldi and M.Bennewitz

Vectors

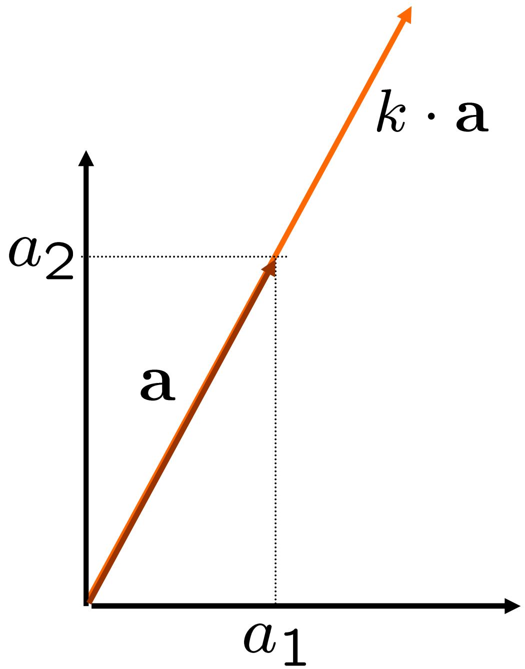
- Arrays of numbers
- They represent a point in a *n* dimensional space

$$\begin{pmatrix} a_1 \end{pmatrix} \quad \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \quad \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}$$



Vectors: Scalar Product

- Scalar-Vector Product $k \cdot \mathbf{a}$
- Changes the length of the vector, but **not** its direction

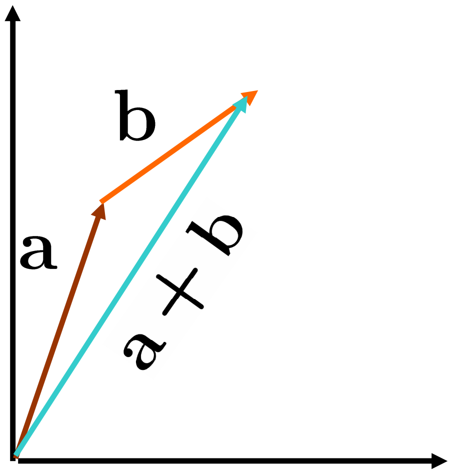


Vectors: Sum

- 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.

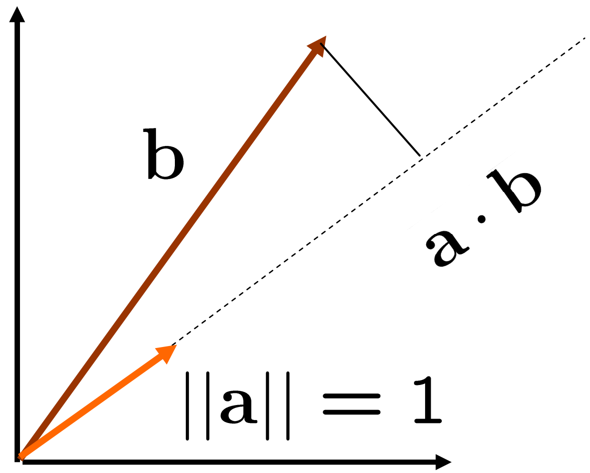


Vectors: Dot Product

- Inner product of vectors (is a scalar)

$$\mathbf{a} \cdot \mathbf{b} = \mathbf{b} \cdot \mathbf{a} = \sum_i a_i \cdot b_i$$

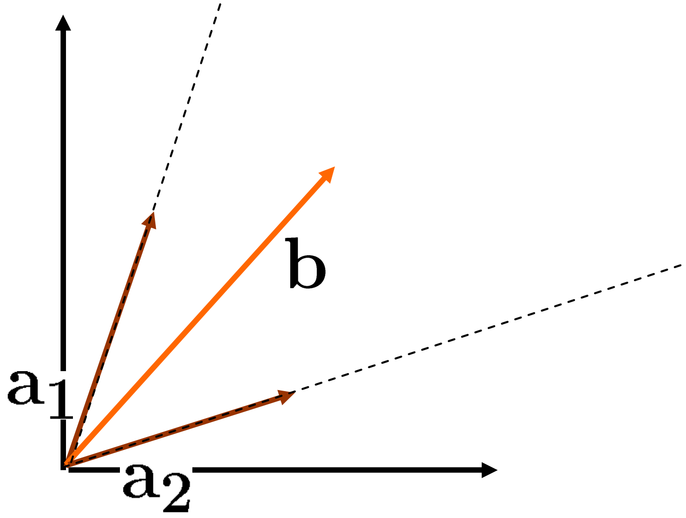
- If one of the two vectors $\mathbf{a}$ has $||\mathbf{a}|| = 1$ the inner product $\mathbf{a} \cdot \mathbf{b}$ returns the length of the projection of $\mathbf{b}$ along the direction of $\mathbf{a}$



- If $\mathbf{a} \cdot \mathbf{b} = 0$ the two vectors are **orthogonal**

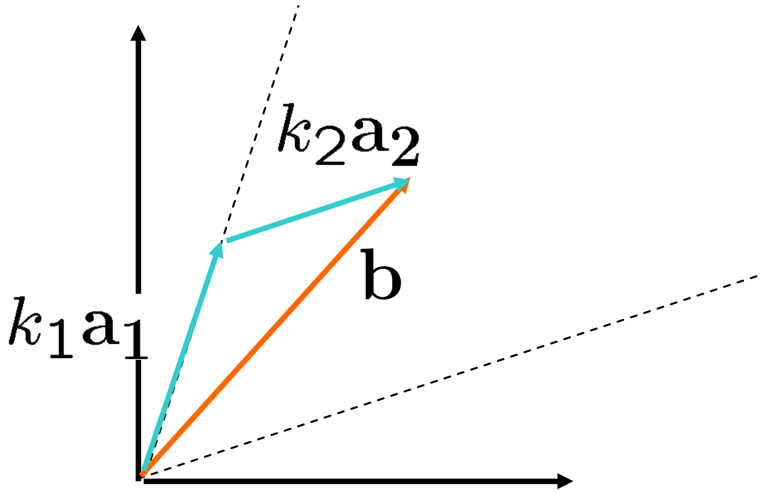
Vectors: Linear (In)Dependence

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



Vectors: Linear (In)Dependence

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



Matrices

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

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

- Note: a d-dimensional vector is equivalent to a dx1 matrix

Matrices as Collections of Vectors

- Column vectors

$$\begin{pmatrix} \mathbf{a}_{*1} & \mathbf{a}_{*2} & \cdots & \mathbf{a}_{*m} \end{pmatrix}$$

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

Matrices as Collections of Vectors

- Row Vectors

$$\mathbf{A} = \begin{pmatrix} \boxed{a_{11} \quad a_{12} \quad \cdots \quad a_{1m}} \\ \boxed{a_{21} \quad a_{22} \quad \cdots \quad a_{2m}} \\ \vdots \\ \boxed{a_{n1} \quad a_{n2} \quad \cdots \quad a_{nm}} \end{pmatrix} \begin{pmatrix} \mathbf{a}_{1*}^T \\ \mathbf{a}_{2*}^T \\ \vdots \\ \mathbf{a}_{*n}^T \end{pmatrix}$$

Matrices Operations

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

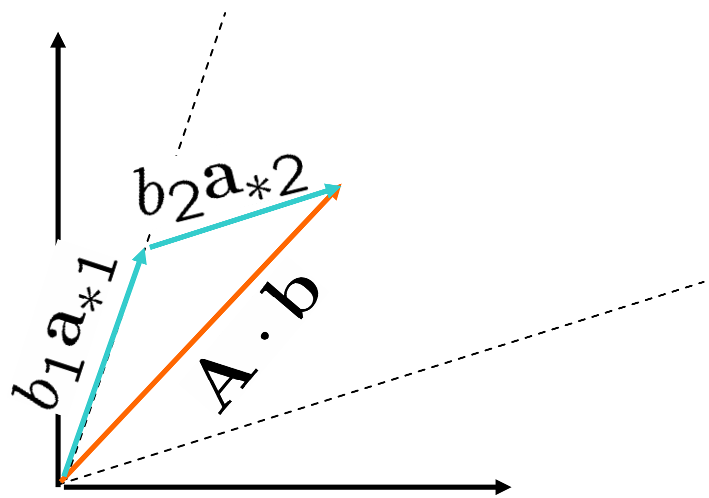
Matrix Vector Product

- The *i* component of $\mathbf{A} \cdot \mathbf{b}$ is the dot product $\mathbf{a}_{i*}^T \cdot \mathbf{b}$
- The vector $\mathbf{A} \cdot \mathbf{b}$ is linearly dependent from $\{\mathbf{a}_{*i}\}$ with coefficients $\{b_i\}$

$$\mathbf{A} \cdot \mathbf{b} = \begin{pmatrix} \mathbf{a}_{1*}^T \\ \mathbf{a}_{2*}^T \\ \vdots \\ \mathbf{a}_{n*}^T \end{pmatrix} \cdot \mathbf{b} = \begin{pmatrix} \mathbf{a}_{1*}^T \cdot \mathbf{b} \\ \mathbf{a}_{2*}^T \cdot \mathbf{b} \\ \vdots \\ \mathbf{a}_{n*}^T \cdot \mathbf{b} \end{pmatrix} = \sum_k \mathbf{a}_{*k} \cdot b_k$$

Matrix Vector Product

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



Matrix Vector Product

- Each $a_{i,j}$ can be seen as a linear mixing coefficient that quantifies the contribution to $(\mathbf{A} \cdot \mathbf{b})_j$
- Example: Jacobian of a multi-dimensional function

$$\mathbf{y} = \mathbf{f}(\mathbf{x}) = \begin{pmatrix} \mathsf{f}_1(\mathbf{x}) \\ \mathsf{f}_2(\mathbf{x}) \\ \vdots \\ \mathsf{f}_n(\mathbf{x}) \end{pmatrix} \quad \mathbf{J}_f = \begin{pmatrix} \frac{d\mathsf{f}_1}{dx_1} & \frac{d\mathsf{f}_1}{dx_2} & \cdots & \frac{d\mathsf{f}_1}{dx_m} \\ \frac{d\mathsf{f}_2}{dx_1} & \frac{d\mathsf{f}_2}{dx_2} & \cdots & \frac{d\mathsf{f}_2}{dx_m} \\ \vdots & \ddots & & \\ \frac{d\mathsf{f}_n}{dx_1} & \frac{d\mathsf{f}_n}{dx_2} & \cdots & \frac{d\mathsf{f}_n}{dx_m} \end{pmatrix}$$

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***

$$
\begin{aligned}
\mathbf{C} &= \mathbf{A} \cdot \mathbf{B} \\
&= \begin{pmatrix}
\mathbf{a}_{1*}^T \cdot \mathbf{b}_{*1} & \mathbf{a}_{1*}^T \cdot \mathbf{b}_{*2} & \cdots & \mathbf{a}_{1*}^T \cdot \mathbf{b}_{*m} \\
\mathbf{a}_{2*}^T \cdot \mathbf{b}_{*1} & \mathbf{a}_{2*}^T \cdot \mathbf{b}_{*2} & \cdots & \mathbf{a}_{2*}^T \cdot \mathbf{b}_{*m} \\
\vdots & & & \\
\mathbf{a}_{n*}^T \cdot \mathbf{b}_{*1} & \mathbf{a}_{n*}^T \cdot \mathbf{b}_{*2} & \cdots & \mathbf{a}_{n*}^T \cdot \mathbf{b}_{*m}
\end{pmatrix} \\
&= \begin{pmatrix} \mathbf{A} \cdot \mathbf{b}_{*1} & \mathbf{A} \cdot \mathbf{b}_{*2} & \ldots \mathbf{A} \cdot \mathbf{b}_{*m} \end{pmatrix}
\end{aligned}
$$

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.

$$\begin{aligned}
\mathbf{C} &= \mathbf{A}\cdot\mathbf{B} \\
&= \left(\ \mathbf{A}\cdot\mathbf{b}_{*1}\ \ \mathbf{A}\cdot\mathbf{b}_{*2}\ \ \ldots\mathbf{A}\cdot\mathbf{b}_{*m}\ \right) \\
\mathbf{c}_{*i} &= \mathbf{A}\cdot\mathbf{b}_{*i}
\end{aligned}$$

Linear Systems

$$\mathbf{Ax} = \mathbf{b}$$

- 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)
    - …
  - 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':****m'xn* and ***b'****:m'x1* and rank **A'** = *min(m',n)*
  - The system might be either over-constrained *(m′>n)* or under-constrained *(m′<n)*

Over-constrained Systems

- An **over-constrained** does not admit an exact solution however if *rank* ***A'*** *=cols(**A**)* one may find a minimum norm solution by closed form pseudo inversion

$$\mathbf{x} = \underset{\mathbf{x}}{\operatorname{argmin}} ||\mathbf{A}'\mathbf{x} - \mathbf{b}'|| = (\mathbf{A}'^T \mathbf{A}')^{-1} \mathbf{A}'^T \mathbf{b}'$$

Linear Systems

- 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 system admits infinite solutions. The degree of infinity is *cols*(***A'***) - *rows*(***A'***)

Matrix Inversion

$$\mathbf{A} \cdot \mathbf{B} = \mathbf{I}$$

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

$$\mathbf{A} \cdot \mathbf{a}^{-1}{}_{*i} = \mathbf{i}_{*i}$$

This is the i^{th} column of the identity matrix

Trace

- Only defined for **square matrices**
- **Sum** of the elements on the main diagonal, that is

$$\mathrm{tr}(A) = a_{11} + a_{22} + \cdots + a_{nn} = \sum_{i=1}^{n} a_{ii}$$

- It is a linear operator with the following properties
  - Additivity: $\mathrm{tr}(A+B) = \mathrm{tr}(A) + \mathrm{tr}(B)$
  - Homogeneity: $\mathrm{tr}(c \cdot A) = c \cdot \mathrm{tr}(A)$
  - Pairwise commutative: $\mathrm{tr}(AB) = \mathrm{tr}(BA), \quad \mathrm{tr}(ABC) \neq \mathrm{tr}(ACB)$

- Trace is similarity invariant $\mathrm{tr}(P^{-1}AP) = \mathrm{tr}((AP^{-1})P) = \mathrm{tr}(A)$

- Trace is transpose invariant $\mathrm{tr}(A) = \mathrm{tr}(A^T)$

- Given two vectors ***a*** and ***b***, $tr(\mathbf{a}^T \mathbf{b}) = tr(\mathbf{a}\,\mathbf{b}^T)$

Rank

- **Maximum** number of linearly independent rows (columns)
- Dimension of the **image** of the transformation $f(\mathbf{x}) = A\mathbf{x}$
- When A is $m \times n$ we have
  - $\operatorname{rank}(A) \geq 0$ and the equality holds iff A is the null matrix
  - $\operatorname{rank}(A) \leq \min(m, n)$
  - $f(\mathbf{x})$ is **injective** iff $\operatorname{rank}(A) = n$
  - $f(\mathbf{x})$ is **surjective** iff $\operatorname{rank}(A) = m$
  - if $m = n$, $f(\mathbf{x})$ is **bijective** and A is **invertible** iff $\operatorname{rank}(A) = n$
- Computation of the rank is done by
  - Gaussian elimination on the matrix
  - Counting the number of non-zero rows

Determinant

- Only defined for **square matrices**
- Remember? $\mathbf{A} \cdot \mathbf{A}^{-1} = \mathbf{I}$ if and only if $det(\mathbf{A}) \neq 0$
- For 2×2 matrices:

  Let $\mathbf{A} = [a_{ij}]$ and $|\mathbf{A}| = det(\mathbf{A})$, then

$$\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11} \cdot a_{22} - a_{12} \cdot a_{21}$$

- For 3×3 matrices the Sarrus rule holds:

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

$$-a_{11}a_{23}a_{32} - a_{12}a_{21}a_{33} - a_{13}a_{22}a_{11}$$

Determinant

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

Let $\mathbf{A}_{ij}$ be the submatrix obtained from $\mathbf{A}$ by deleting the *i-th* row and the *j-th* column

$$\begin{bmatrix} 1 & 2 & 5 & 0 \\ 2 & 3 & 4 & -1 \\ -5 & 8 & 0 & 0 \\ 0 & 4 & -2 & 0 \end{bmatrix} \Rightarrow \mathbf{A}_{23} = \begin{bmatrix} 1 & 5 & 0 \\ 2 & 4 & -1 \\ 0 & -2 & 0 \end{bmatrix}$$

Rewrite determinant for 3×3 matrices:

$$\begin{aligned} det(\mathbf{A}_{3\times 3}) &= a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{21}a_{32} \\ &\quad -a_{11}a_{23}a_{32} - a_{12}a_{21}a_{33} - a_{13}a_{22}a_{11} \\ &= a_{11} \cdot det(\mathbf{A}_{11}) - a_{12} \cdot det(\mathbf{A}_{12}) + a_{13} \cdot det(\mathbf{A}_{13}) \end{aligned}$$

Determinant

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

$$
\begin{aligned}
det(\mathbf{A}) &= a_{11}det(\mathbf{A}_{11}) - a_{12}det(\mathbf{A}_{12}) + ... + (-1)^{1+n}a_{1n}det(\mathbf{A}_{1n}) \\
&= \sum_{j=1}^{n}(-1)^{1+j}a_{1j}det(\mathbf{A}_{1j})
\end{aligned}
$$

Let $\mathbf{C}_{ij} = (-1)^{i+j}det(\mathbf{A}_{ij})$ be the *(i,j)*-cofactor, then

$$
\begin{aligned}
det(\mathbf{A}) &= a_{11}\mathbf{C}_{11} + a_{12}\mathbf{C}_{12} + ... + a_{1n}\mathbf{C}_{1n} \\
&= \sum_{j=1}^{n}a_{1j}\mathbf{C}_{1j}
\end{aligned}
$$

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 x 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:

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

  Because for **triangular matrices** the determinant is the product of diagonal elements

Determinant: Properties

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

- **Transpose**: $det(\mathbf{A}^T) = det(\mathbf{A})$

- **Multiplication**: $det(\mathbf{A} \cdot \mathbf{B}) = det(\mathbf{A}) \cdot det(\mathbf{B})$

- Does **not** apply to addition! $det(\mathbf{A} + \mathbf{B}) \neq det(\mathbf{A}) + det(\mathbf{B})$

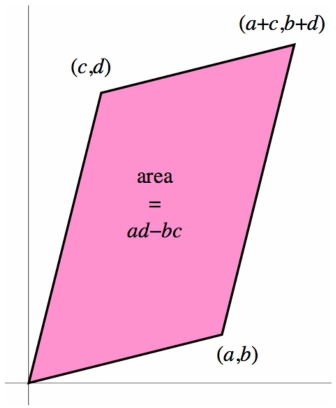
Determinant: Applications

- Find **the inverse** $\mathbf{A}^{-1}$ using Cramer's rule $\mathbf{A}^{-1} = \dfrac{\mathrm{adj}(\mathbf{A})}{det(\mathbf{A})}$
  with $\mathrm{adj}(\mathbf{A})$ being the adjugate of $\mathbf{A}$

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

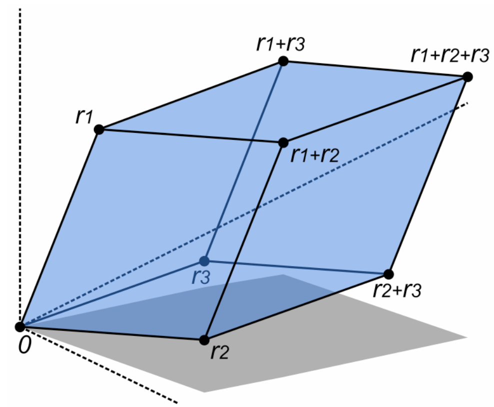
- **Area** and **Volume:** $\mathrm{area} = |det(\mathbf{A})|$

$$\mathbf{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$



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

(r_i is i-th row)



Orthonormal matrix

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

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

- 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)

$$1 = det(I) = det(Q^TQ) = det(Q)det(Q^T) = det(Q)^2$$

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

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix} \quad R_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix}$$

- IMPORTANT: Rotations are **not commutative**

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

$$R_y(\frac{\pi}{4}) \cdot R_x(\frac{\pi}{4}) = \begin{bmatrix} 0.707 & -0.5 & -0.5 \\ 0 & 0.707 & -0.707 \\ 0.707 & 0.5 & 0.5 \end{bmatrix}, \quad R_y(\frac{\pi}{4}) \cdot R_x(\frac{\pi}{4}) \cdot \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} -1.793 \\ 0.707 \\ 3.207 \end{bmatrix}$$

Matrices to represent Affine Transformations

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

Translation Vector

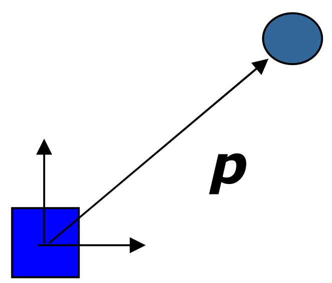
$$\mathbf{A} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{pmatrix} \quad \mathbf{A}^{-1} = \begin{pmatrix} \mathbf{R}^T & -\mathbf{R}^T\mathbf{t} \\ 0 & 1 \end{pmatrix} \quad \mathbf{p} = \begin{pmatrix} \mathbf{t} \\ 1 \end{pmatrix}$$

Rotation Matrix

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

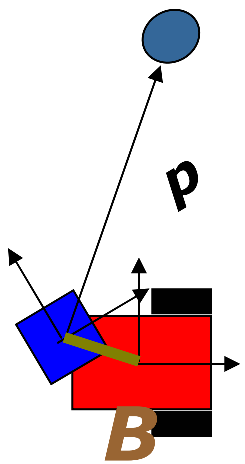
Combining Transformations

- 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

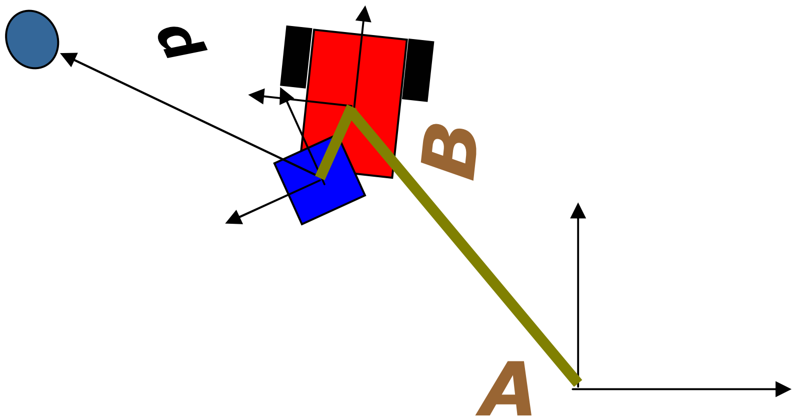
- A simple interpretation: chaining of transformations (represented ad homogeneous matrices)
  - Matrix ***A*** represents the pose of a **robot** in the space
  - Matrix ***B*** represents the position of a sensor on the robot
  - The **sensor** perceives an **object** at a given location ***p***, in 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 ad 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 A is **symmetric** if $A = A^T$, e.g. $\begin{bmatrix} 1 & 4 & -2 \\ 4 & -1 & 3 \\ -2 & 3 & 5 \end{bmatrix}$

- A matrix A is **skew-symmetric** if $A = -A^T$, e.g. $\begin{bmatrix} 0 & 4 & -2 \\ -4 & 0 & 3 \\ 2 & -3 & 0 \end{bmatrix}$

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

Positive definite matrix

- The analogous of positive number
- Definition $M > 0$ iff $z^T M z > 0 \forall z \neq 0$
- Examples
  - $$M_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} z_1 & z_2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = z_1^2 + z_2^2 > 0$$
  - $$M_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} z_1 & z_2 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = 2z_1 z_2 < 0, \; z_1 = -z_2$$

Positive definite matrix

- Properties
  - **Invertible**, with positive definite inverse
  - All real **eigenvalues** > 0
  - **Trace** is > 0

  - **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
    - $M + N > 0$

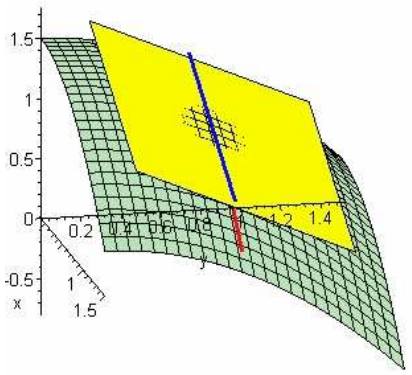
Jacobian Matrix

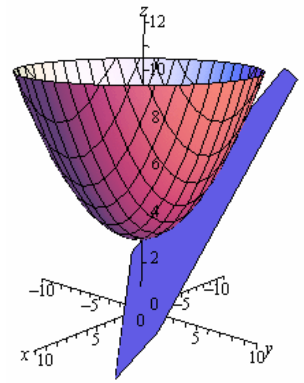
- It's a **non-square matrix** $n \times m$ in general
- Suppose you have a vector-valued function $f(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \\ \vdots \\ f_m(\mathbf{x}) \end{bmatrix}$
- Then, the **Jacobian matrix** is defined as

$$\mathbf{F_x} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

Jacobian Matrix

- 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**

$$\mathbf{C}_{out} = \mathbf{F} \cdot \mathbf{C}_{in} \cdot \mathbf{F}^T$$

$\rightarrow$ See later in the course

Quadratic Forms

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

$$\begin{aligned} f(\mathbf{x}) &= \sum_{i,j} a_{ij} x_i x_j + \sum_i b_i x_i + c \\ &= \mathbf{x}^T \mathbf{A} \mathbf{x} + \mathbf{b} \mathbf{x} + c \end{aligned}$$

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

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}} f(\mathbf{x})$$

Quadratic Forms

- How can we use the matrix properties to quickly compute a solution to this minimization problem?

$$\hat{\mathbf{x}} = \operatorname*{argmin}_{\mathbf{x}} f(\mathbf{x})$$

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

$$\begin{aligned} f(\mathbf{x}) &= \mathbf{x}^T\mathbf{A}\mathbf{x} + \mathbf{b}\mathbf{x} + c \\ f'(\mathbf{x}) &= \mathbf{A}^T\mathbf{x} + \mathbf{A}\mathbf{x} + \mathbf{b} \end{aligned}$$

Quadratic Forms

- The minimum of $f(\mathrm{x}) = \mathrm{x}^T\mathrm{Ax} + \mathrm{bx} + c$ is where its derivative is set to 0

$$0 = \mathbf{A}^T\mathbf{x} + \mathbf{Ax} + \mathbf{b}$$

- Thus we can solve the system

$$(\mathbf{A}^T + \mathbf{A})^T\mathbf{x} = -\mathbf{b}$$

- If the matrix is symmetric, the system becomes

$$2\mathbf{Ax} = -\mathbf{b}$$

- Solving that, leads to the minimum