Introduction to Mobile Robotics

Compact Course on Linear Algebra

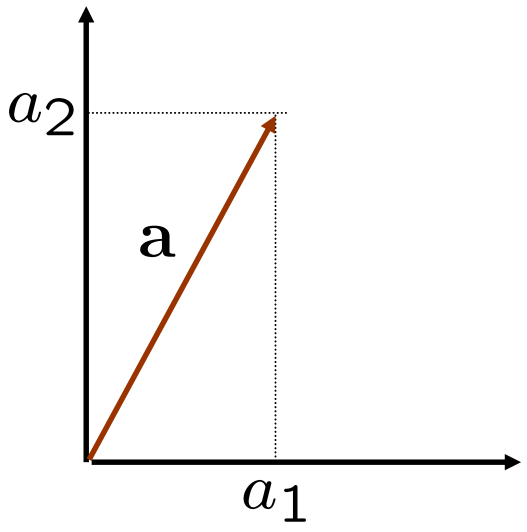
Wolfram Burgard, Michael Krawez

UTN

Vectors

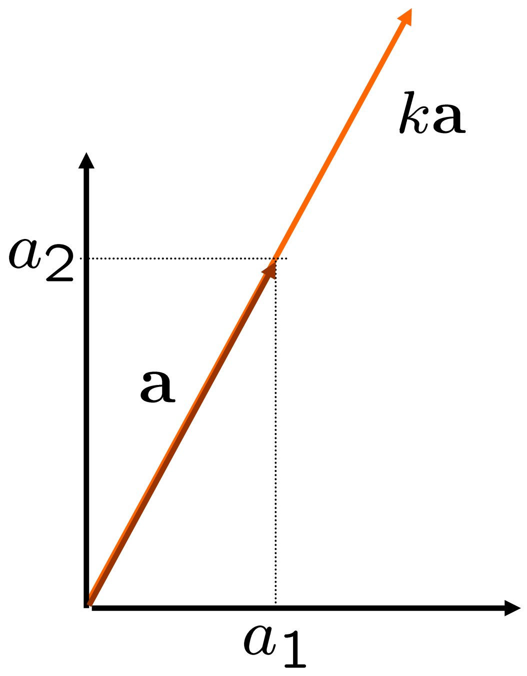
- Arrays of numbers
- Vectors represent a point in an *n*-dimensional space

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



Vectors: Scalar Product

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

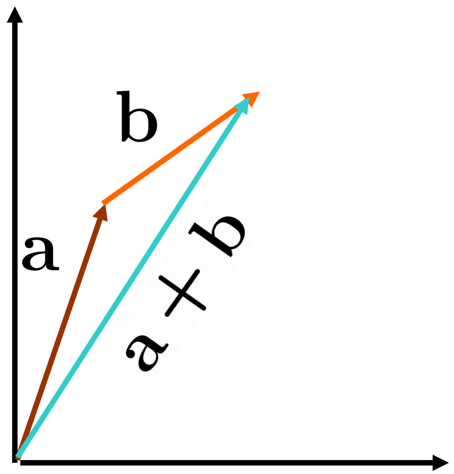


Vectors: Sum

- Sum of vectors (is commutative)

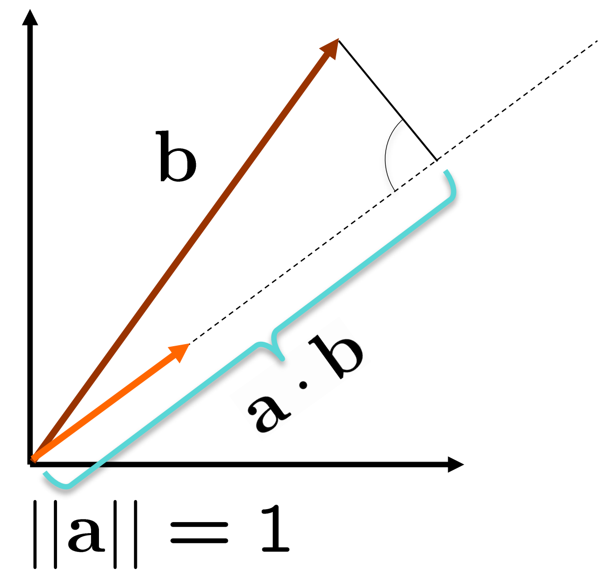
$$\begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} + \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}$$

- Can be visualized as “chaining” the vectors.



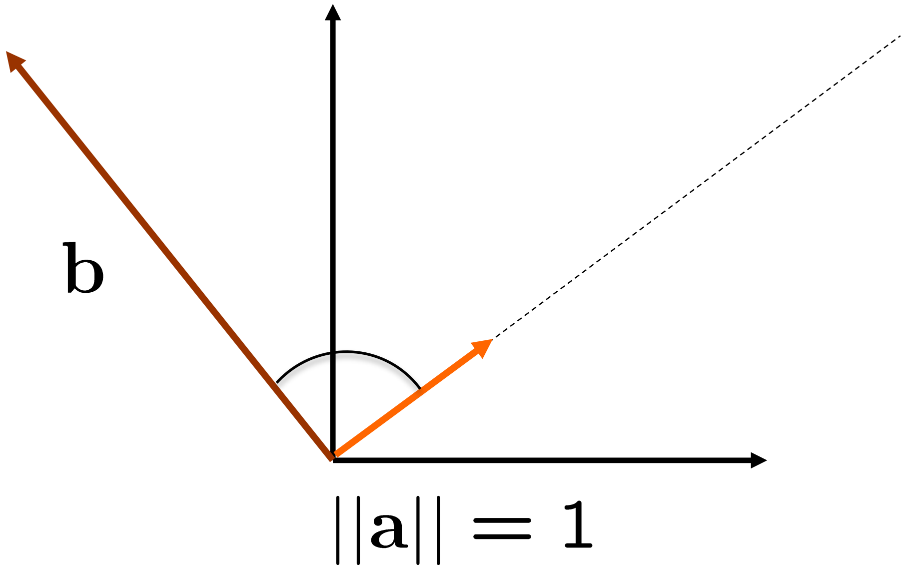
Vectors: Dot Product

- Inner product of vectors (is a scalar) $\mathbf{a} \cdot \mathbf{b} = \mathbf{b} \cdot \mathbf{a} = \sum_i a_i b_i$
- If one of the two vectors, e.g., $\mathbf{a}$, 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 $\mathbf{a}$



Vectors: Dot Product

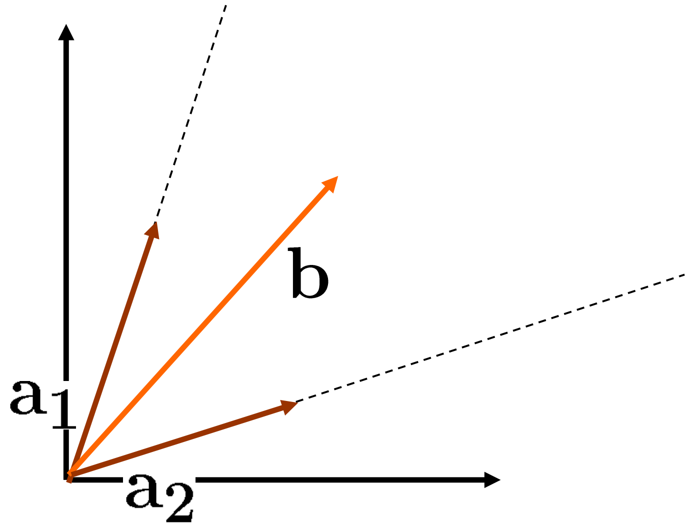
- Inner product of vectors (is a scalar) $\mathbf{a} \cdot \mathbf{b} = \mathbf{b} \cdot \mathbf{a} = \sum_i a_i b_i$
- If one of the two vectors, e.g., $\mathbf{a}$, has $||\mathbf{a}|| = 1$, the inner product $\mathbf{a} \cdot \mathbf{b}$ returns the length of the projection of $\mathbf{b}$ along the direction of $\mathbf{a}$



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

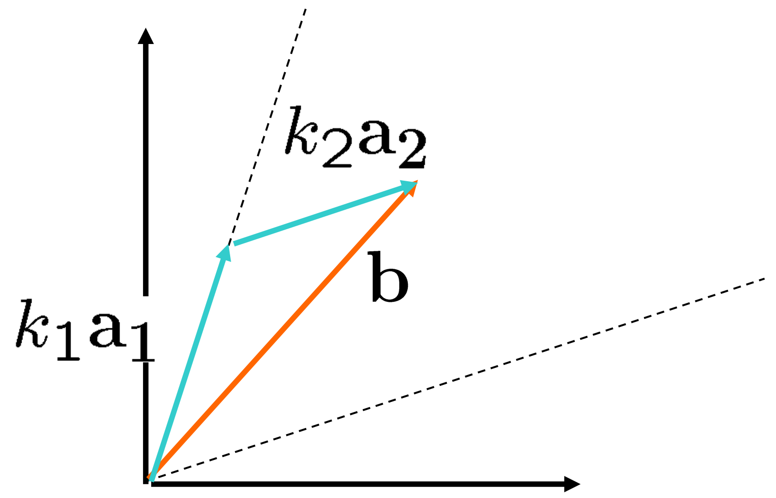
Vectors: Linear (In)Dependence

- A vector $\mathbf{b}$ is **linearly dependent** from $\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n\}$ if $\mathbf{b} = \sum_i k_i \mathbf{a}_i$
- In other words, if $\mathbf{b}$ can be obtained by summing up the $\mathbf{a}_i$ properly scaled
- If there exists no $\{k_i\}$ such that $\mathbf{b} = \sum_i k_i \mathbf{a}_i$ then $\mathbf{b}$ is independent from $\{\mathbf{a}_i\}$



Vectors: Linear (In)Dependence

- A vector $\mathbf{b}$ is **linearly dependent** from $\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n\}$ 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 exists 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

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & & & \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{pmatrix} \qquad A : \underset{\text{rows}}{\underset{\uparrow}{n}} \times \underset{\text{columns}}{\underset{\uparrow}{m}}$$

- **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

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

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

Matrices as Collections of Vectors

- Row vectors

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

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

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

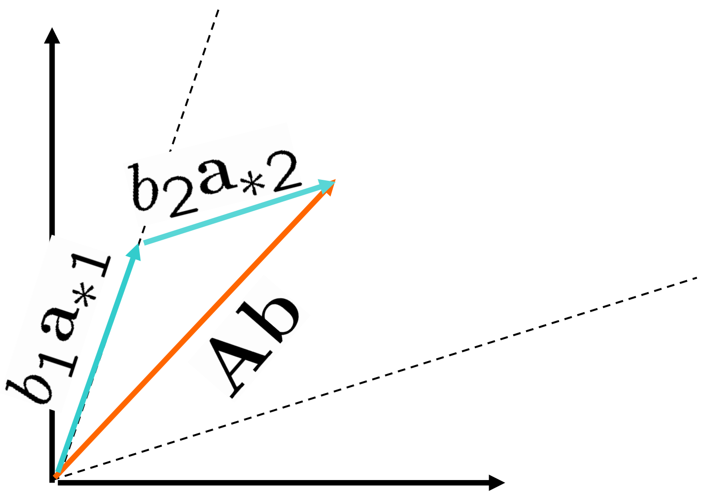
row vectors

column vectors

Matrix Vector Product

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

column vectors



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***

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

column vectors

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

$$
\begin{aligned}
\mathrm{C} &= \mathbf{A}\mathbf{B} \\
&= \begin{pmatrix} \mathbf{A}\mathbf{b}_{*1} & \mathbf{A}\mathbf{b}_{*2} & \ldots \mathbf{A}\mathbf{b}_{*m} \end{pmatrix} \\
\mathbf{c}_{*i} &= \mathbf{A}\mathbf{b}_{*i}
\end{aligned}
$$

column vectors

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
  - $\mathrm{rank}(A) \geq 0$ and the equality holds iff A is the null matrix
  - $\mathrm{rank}(A) \leq \min(m, n)$

- Computation of the rank is done by
  - Gaussian elimination on the matrix and then
  - Counting the number of non-zero rows

Inverse

$$\mathrm{AB} = \mathrm{I}$$

- If A is a square matrix of full rank, then there is a unique matrix $\boldsymbol{B}=\boldsymbol{A}^{-1}$ such that $\boldsymbol{AB}=\boldsymbol{BA}=\boldsymbol{I}$ holds
- The i^{th} row of $\boldsymbol{A}$ is and the j^{th} column of $\boldsymbol{A}^{-1}$ are:
  - orthogonal (if $i \neq j$)
  - or their dot product is 1 (if $i = j$)

Matrix Inversion

$$\mathbf{AB} = \mathbf{I}$$

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

$$\mathbf{Aa^{-1}}_{*i} = \mathbf{i}_{*i}$$

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

Determinant (det)

- Only defined for **square matrices**
- The inverse of $\mathbf{A}$ exists 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

$$\begin{bmatrix} 1 & 2 & 5 & 0 \\ 2 & 3 & 4 & -1 \\ -5 & 8 & 0 & 0 \\ 0 & 4 & -2 & 0 \end{bmatrix} \Rightarrow \mathbf{A}_{32} = \begin{bmatrix} 1 & 5 & 0 \\ 2 & 4 & -1 \\ 0 & -2 & 0 \end{bmatrix}$$

  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 modern supercomputer would take **a year**.

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

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

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

Determinant: Properties

- **Row operations** ($\mathbf{A}$ is still a $n \times n$ square matrix)
  - If $\mathbf{B}$ results from $\mathbf{A}$ by swapping 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 $det(\mathbf{B}) = det(\mathbf{A})$

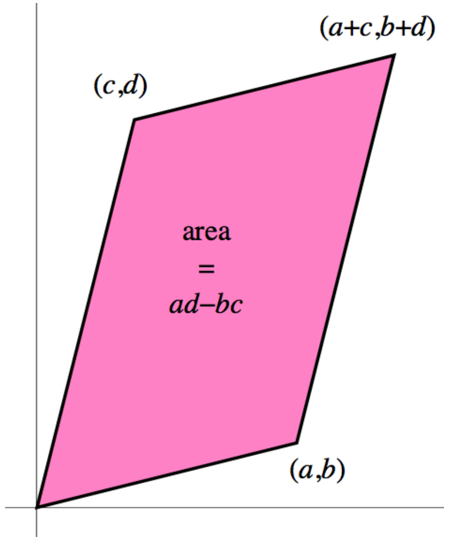
- **Transpose**: $det(\mathbf{A}^T) = det(\mathbf{A})$
- **Multiplication**: $det(\mathbf{A} \cdot \mathbf{B}) = det(\mathbf{A}) \cdot det(\mathbf{B})$
- Does **not** apply to addition! $det(\mathbf{A} + \mathbf{B}) \neq det(\mathbf{A}) + det(\mathbf{B})$

Determinant: Applications

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

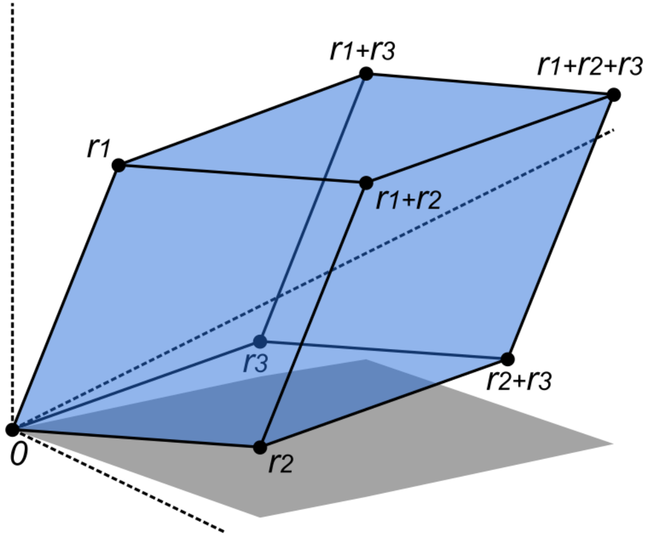
- **Area** and **Volume:** $\text{area} = |det(\mathbf{A})|$

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



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

(r_i is *i-th* row)



Orthogonal Matrix

- A matrix Q is **orthogonal** 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

- Some properties of orthogonal matrices:
  - 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**

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

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

Matrices to Represent Affine Transformations

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

Translation Vector

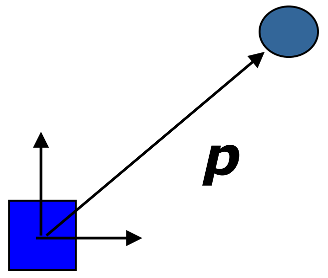
$$\mathbf{A} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{pmatrix} \quad \mathbf{A}^{-1} = \begin{pmatrix} \mathbf{R}^T & -\mathbf{R}^T\mathbf{t} \\ \mathbf{0} & 1 \end{pmatrix} \quad \mathbf{p} = \begin{pmatrix} \mathbf{t} \\ 1 \end{pmatrix}$$

Rotation Matrix

- Takes naturally into account the non-commutativity of the transformations
- Homogeneous coordinates

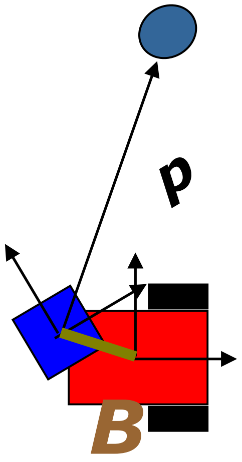
Combining Transformations

- 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

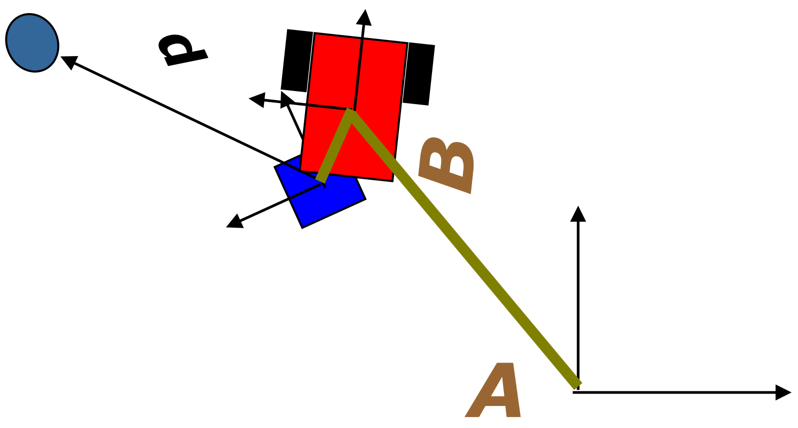
- A simple interpretation: chaining of transformations (represented as homogeneous matrices)
  - Matrix ***A*** represents the pose of a **robot** in the space
  - Matrix ***B*** represents the position of a sensor on the robot
  - The **sensor** perceives an **object** at a given location ***p***, in 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)

$$\mathrm{Ax} = \mathrm{b}$$

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

Gaussian Elimination

A method to solve systems of linear equations.

Example for three variables:

$$
\begin{array}{ccccccc}
a_{11}x_1 & + & a_{12}x_2 & + & a_{13}x_3 & = & b_1 \\
a_{21}x_1 & + & a_{22}x_2 & + & a_{23}x_3 & = & b_2 \\
a_{31}x_1 & + & a_{32}x_2 & + & a_{33}x_3 & = & b_3
\end{array}
$$

We want to transform this to

$$
\begin{array}{ccccccc}
\tilde{a}_{11}x_1 & + & \tilde{a}_{12}x_2 & + & \tilde{a}_{13}x_3 & = & \tilde{b}_1 \\
 & & \tilde{a}_{22}x_2 & + & \tilde{a}_{23}x_3 & = & \tilde{b}_2 \\
 & & & & \tilde{a}_{33}x_3 & = & \tilde{b}_3.
\end{array}
$$

Gaussian Elimination

Written as an extended coefficient matrix:

$$\left(\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ a_{31} & a_{32} & a_{33} & b_3 \end{array}\right) \rightarrow \left(\begin{array}{ccc|c} \tilde{a}_{11} & \tilde{a}_{12} & \tilde{a}_{13} & \tilde{b}_1 \\ 0 & \tilde{a}_{22} & \tilde{a}_{23} & \tilde{b}_2 \\ 0 & 0 & \tilde{a}_{33} & \tilde{b}_3 \end{array}\right)$$

To reach this form, we only need two elementary row operations:

- Add to one row a scalar multiple of another.
- Swap the positions of two rows.

Another commonly used term for Gaussian Elimination is *row reduction.*

Linear Systems (2)

$$\mathrm{Ax} = \mathrm{b}$$

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 $\boldsymbol{A'x=b'}$ the reduced system with $\boldsymbol{A'}: n' x\ m$ and $\boldsymbol{b'}: n' x\ 1$ and rank $\mathbf{A'} = min(n',m)$

  rows columns
- The system might be either over-constrained $(n'>m)$ or under-constrained $(n'<m)$

Over-Constrained Systems

- "More (independent) equations than variables"
- An over-constrained system does not admit an **exact solution**
- However, if *rank* ***A'*** = *cols(**A**)* one often computes a **minimum norm solution**

$$\mathbf{x} = \operatorname*{argmin}_{\mathbf{x}} ||\mathbf{A}'\mathbf{x} - \mathbf{b}'||$$

Note: rank = Maximum number of linearly independent rows/columns

Under-Constrained Systems

- "More variables than (independent) equations"
- The system is **under-constrained** if the number of linearly independent rows of $\boldsymbol{A'}$ is smaller than the dimension of $\boldsymbol{b'}$
- An under-constrained system admits infinitely many solutions
- The degree of these solutions is $cols(\boldsymbol{A'}) - rows(\boldsymbol{A'})$

Jacobian Matrix

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

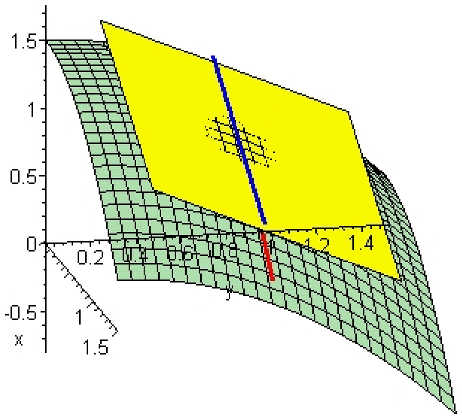
$$f(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \\ \vdots \\ f_m(\mathbf{x}) \end{bmatrix}$$

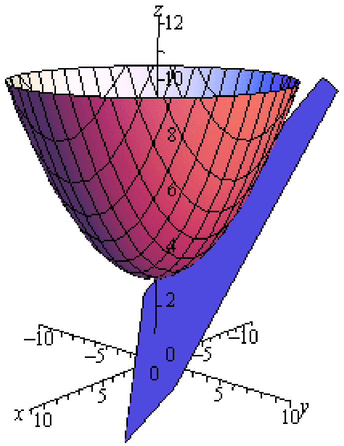
- The **Jacobian matrix** is defined as

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

Jacobian Matrix

- It 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.

Just use your favorite search engine and search for “matrix cookbook” to find it.