CS 565 – Computer Vision

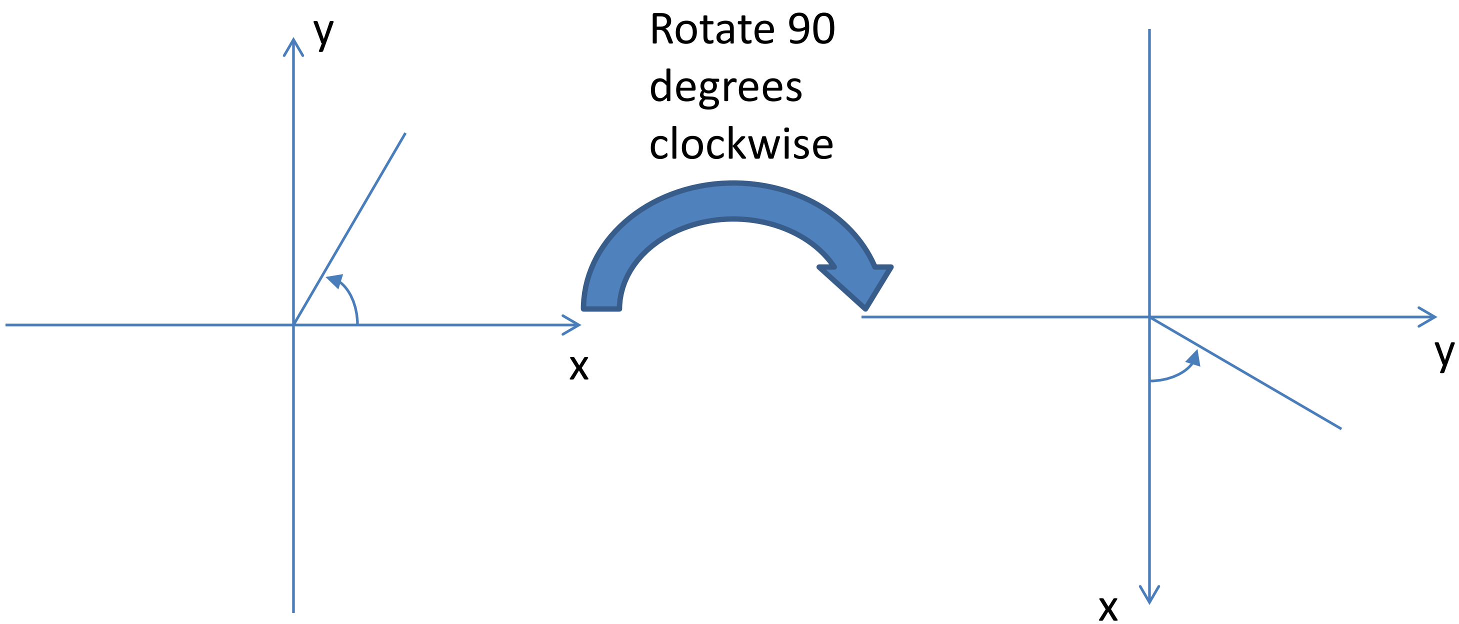
Nazar Khan
PUCIT

Lecture 2: Mathematical Background

Mathematical Background

1. Cartesian vs. Image axes
2. Taylor series expansion
3. Matrix and Vector calculus
4. Eigenvectors
5. Constrained optimisation
6. SVD

Cartesian vs. Image axes



Cartesian axes

- Positive x-axis goes from left to right
- Positive y-axis goes upwards
- Angle measured counter-clockwise from positive-x-axis

Image axes

- Positive x-axis goes downwards
- Positive y-axis goes from left to right
- Angle measured counter-clockwise from positive-x-axis

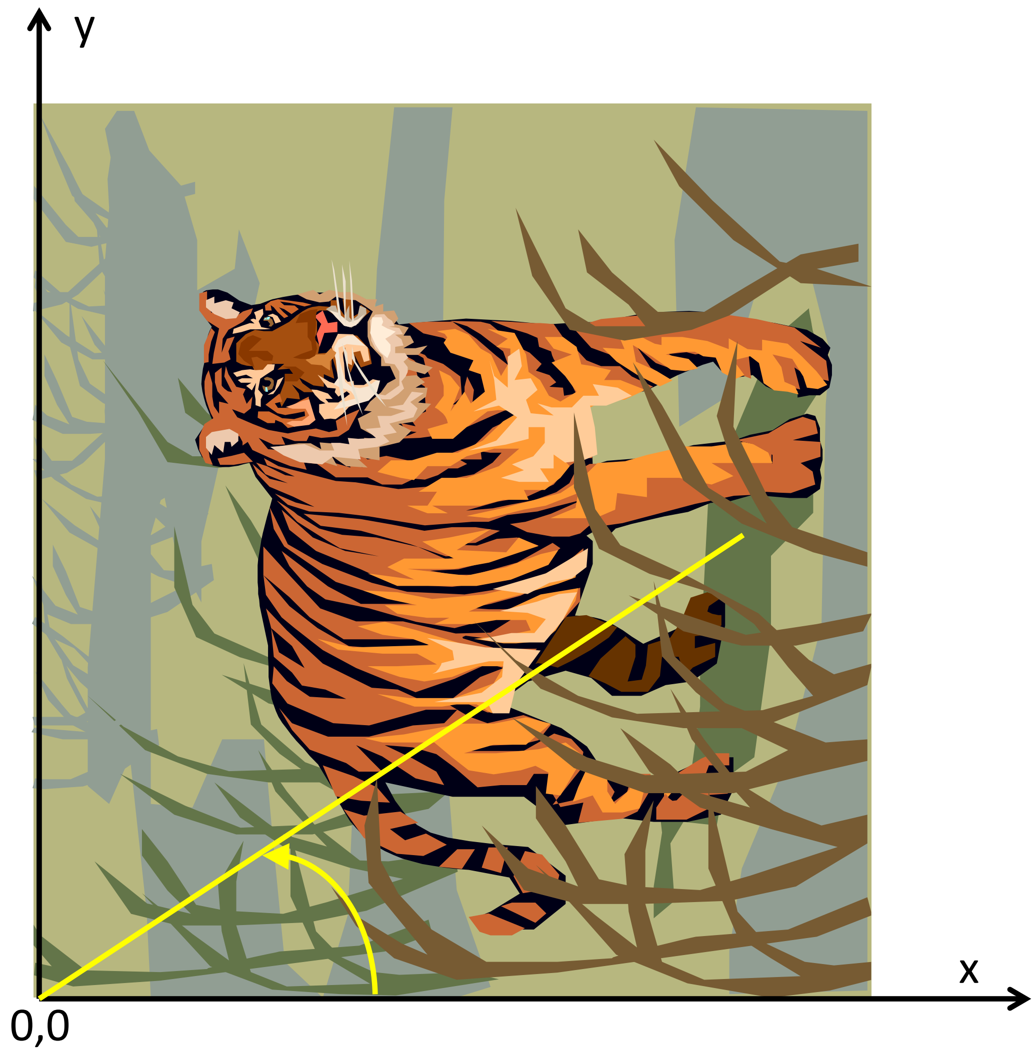
Cartesian vs. Image axes

y-axis goes horizontally left-to-right from top-left corner (y=column)

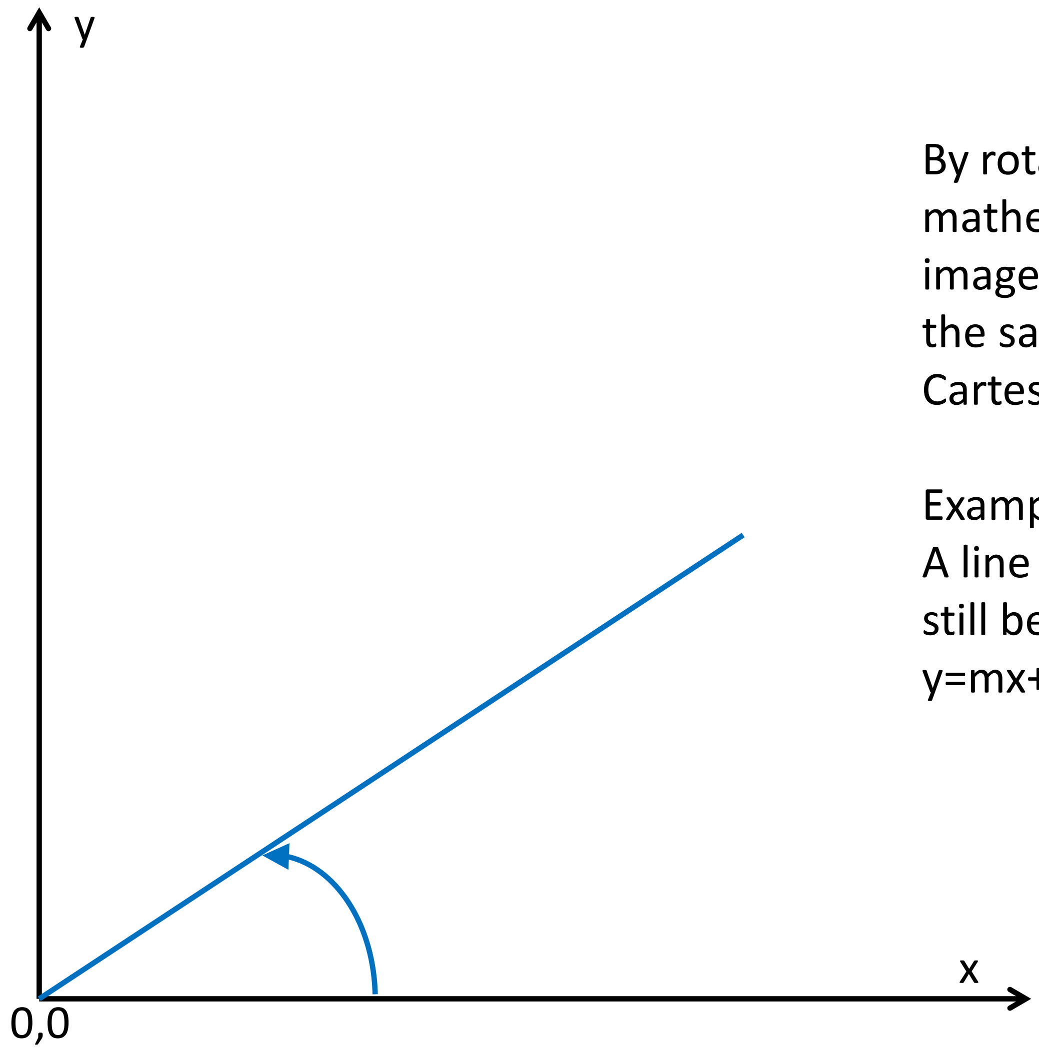
Angle measured in counter-clockwise direction from +ve x-axis

x-axis goes vertically downwards from top-left corner (x=row)

Cartesian vs. Image axes



Cartesian vs. Image axes



By rotating the axis, the mathematics on the image axes will remain the same as for the Cartesian axes.

Example:
A line in the image can still be represented via y=mx+c.

Taylor series expansion

- If values of a function *f(a)* and its derivatives *f'(a), f''(a), …* are known at a value *a*, then we can approximate *f(x)* <u>for *x* close to *a*</u> via the Taylor series expansion:

$$f(x) \approx f(a) + (x-a)\, f'(a)/1! + (x-a)^2\, f''(a)/2! + (x-a)^3\, f'''(a)/3! + O((x-a)^4)$$

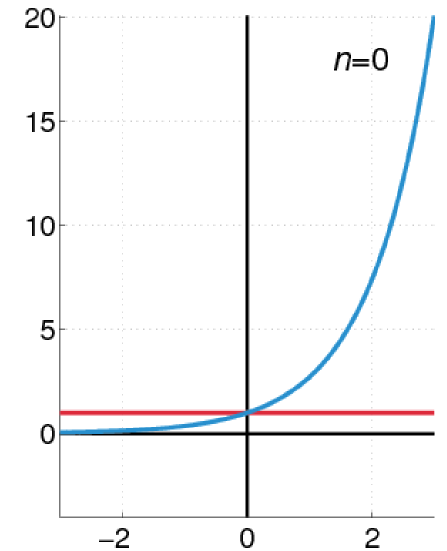
- Examples

  For x around a=0

  - $\sin(x) \approx x - x^3/3! + x^5/5! - x^7/7! + \ldots$
  - $e^x \approx 1 + x^2/2! + x^3/3! + x^4/4! + \ldots$

- Often the first-order Taylor expansion is used

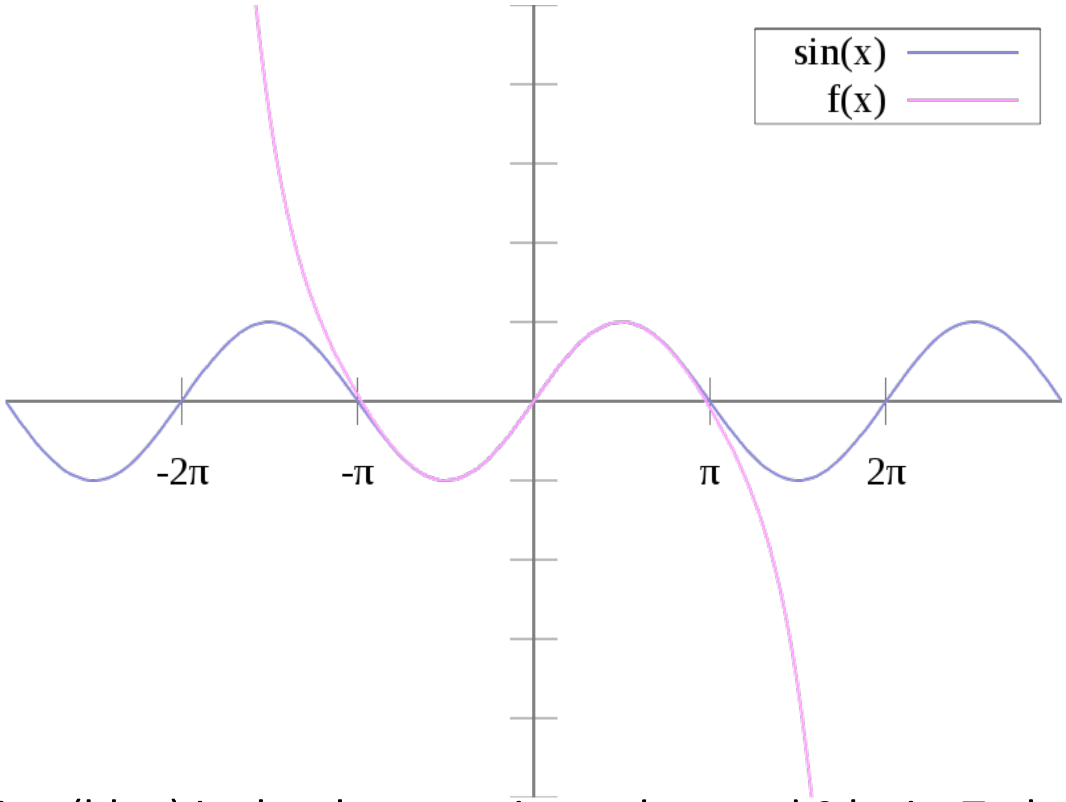
$$f(x) \approx f(a) + (x-a)\, f'(a)/1!$$



The exponential function e^x (in blue), and the sum of the first n+1 terms of its Taylor series at 0 (in red).

Taylor series expansion

- Not very useful for *x* not close to *a*.



The sine function (blue) is closely approximated around 0 by its Taylor polynomial of degree 7 (pink) for a full period centered at the origin. Notice that the approximation becomes poor for $|x-a|>\pi$. Source: https://en.wikipedia.org/wiki/Taylor_series

Matrices and Vectors

- Vectors are denoted by lower-case bold letters like **x**, **y**, **v** etc.
- Matrices are denoted by upper-case bold letters like **M**, **D**, **A** etc.
- A vector $\mathbf{x} \in \mathbb{R}^d$ is by default a column vector $\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}$
- The corresponding row vector is obtained as $\mathbf{x}^T = [x_1\ x_2\ \dots\ x_d]$.

Matrices and Vectors

For vectors $\mathbf{x} \in \mathbb{R}^d$ and $\mathbf{y} \in \mathbb{R}^d$ and $\mathbf{z} \in \mathbb{R}^k$

- Inner product $\mathbf{x}^T\mathbf{y} = x_1y_1 + x_2y_2 + \ldots + x_dy_d$ is a scalar value.
- Also called dot product or scalar product.
- Other representations: $x \cdot y$ and (x,y)
- Represents similarity of vectors.
  - If $\mathbf{x}^T\mathbf{y} = 0$, then $\mathbf{x}$ and $\mathbf{y}$ are orthogonal vectors (in 2D, this means they are perpendicular).

Matrices and Vectors

For vectors $\mathbf{x} \in \mathbb{R}^d$ and $\mathbf{y} \in \mathbb{R}^d$ and $\mathbf{z} \in \mathbb{R}^k$

- Euclidean norm of vector
  $||\mathbf{x}|| = \text{sqrt}(\mathbf{x}^T\mathbf{x}) = \text{sqrt}(x_1x_1 + x_2x_2 + \ldots + x_dx_d)$
  represents the magnitude of the vector.
- <u>Unit vector</u> has norm 1. Also called <u>normalised vector</u>.
- If $||\mathbf{x}||=1$ and $||\mathbf{y}||=1$, and $\mathbf{x}^T\mathbf{y} = 0$, then **x** and **y** are <u>orthonormal</u> vectors.
- <u>Outer-product</u> $\mathbf{x}\mathbf{z}^T$ is a d x k matrix.

Matrix and Vector Calculus

For vectors $\mathbf{x} \in \mathbb{R}^d$, $\mathbf{y} \in \mathbb{R}^d$ and matrix $\mathbf{M} \in \mathbb{R}^{k \times d}$ and scalar function $f(\mathbf{x})$

- $d(\mathbf{y}^T\mathbf{x})/d\mathbf{x} = d(\mathbf{x}^T\mathbf{y})/d\mathbf{x} = \mathbf{y}$
- $d(\mathbf{M}\mathbf{x})/d\mathbf{x} = \mathbf{M}$
- $d(\mathbf{x}^T\mathbf{M}\mathbf{x})/d\mathbf{x} = (\mathbf{M}+\mathbf{M}^T)\mathbf{x}$
- For symmetric $\mathbf{M}$, $d(\mathbf{x}^T\mathbf{M}\mathbf{x})/d\mathbf{x} = \mathbf{2M}\mathbf{x}$
- $d(f(\mathbf{x}))/d\mathbf{x} = \begin{bmatrix} \frac{\partial f(x)}{\partial x_1} \\ \frac{\partial f(x)}{\partial x_2} \\ \vdots \\ \frac{\partial f(x)}{\partial x_d} \end{bmatrix}$

Verify all of these derivatives

Matrix and Vector calculus

For vector $\mathbf{x} \in \mathbb{R}^d$ and vector function $\mathbf{g}(\mathbf{x}) \in \mathbb{R}^k$

$$\mathrm{d}(\mathbf{g}(\mathbf{x}))/\mathrm{d}\mathbf{x} = \begin{bmatrix} \frac{\partial g_1(\mathbf{x})}{\partial x_1} & \frac{\partial g_1(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial g_1(\mathbf{x})}{\partial x_d} \\ \frac{\partial g_2(\mathbf{x})}{\partial x_1} & \frac{\partial g_2(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial g_2(\mathbf{x})}{\partial x_d} \\ \vdots & & \ddots & \\ \frac{\partial g_k(\mathbf{x})}{\partial x_1} & \cdots & & \frac{\partial g_k(\mathbf{x})}{\partial x_d} \end{bmatrix}$$

Matrix and Vector calculus

- The gradient operator d/d**x** is also written as $\nabla_{\mathbf{x}}$ or ∇ when the differentiation variable is implied.
- $\nabla_{\mathbf{x}}(\mathbf{Mx}) = d(\mathbf{Mx})/d\mathbf{x} = \mathbf{M}$ **(Verify this)**
- $\nabla_{\mathbf{x}} = \begin{bmatrix} \dfrac{\partial}{\partial x_1} \\ \dfrac{\partial}{\partial x_2} \\ \vdots \\ \dfrac{\partial}{\partial x_d} \end{bmatrix}$

Matrices as linear operators

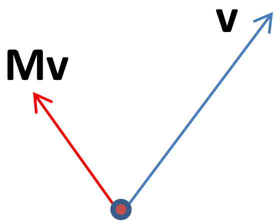
- In a matrix transformation **Mx**, components of **x** are acted upon in a linear fashion.

$$\begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} m_{11}x_1 + m_{12}x_2 \\ m_{12}x_1 + m_{22}x_2 \end{bmatrix}$$

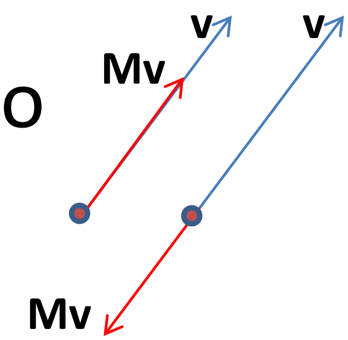
- <u>Every</u> matrix represents a linear transformation.
- <u>Every</u> linear transformation can be represented as a matrix.

Eigenvectors

- Matrix-vector product **Mv**
- When a matrix **M** is multiplied with a vector **v**, the vector is linearly transformed
  - Rotation and/or
  - Scaling



- If **v** is **not rotated** but **only scaled** then it is called an **eigenvector** of **M**.
- **Mv**=λ**v** where λ is the scaling factor (also called the **eigenvalue**).



Constrained optimisation

- For optimising a function f(x) the gradient of f must vanish at the optimiser x*

$$\nabla f|_{x^*} = 0$$

- For optimising a function f(x) subject to some constraint g(x)=0, the gradient of the so-called Lagrange function

$$L(x, \lambda) = f(x)+\lambda g(x)$$

must vanish at the optimiser x*

$$\nabla L(x, \lambda) = \nabla f|_{x^*} + \lambda \nabla g|_{x^*} = 0$$

where λ is the Lagrange (or undetermined) multiplier.

Constrained optimisation

- Quite often, we will need to maximise $\mathbf{x}^T\mathbf{M}\mathbf{x}$ with respect to $\mathbf{x}$ where $\mathbf{M}$ is a symmetric matrix.
  - Trivial solution: $\mathbf{x}=\infty$
- To prevent trivial solution, we must constrain the norm of $\mathbf{x}$. For example, $\mathbf{x}^T\mathbf{x} = 1$.
- Lagrangian becomes $L(\mathbf{x},\lambda)= \mathbf{x}^T\mathbf{M}\mathbf{x} + \lambda(\mathbf{x}^T\mathbf{x}-1)$
- Use $\partial L/\partial \mathbf{x} = 0$ and $\partial L/\partial \lambda = 0$ to solve for optimal $\mathbf{x}^*$. **(H.W. Try this)**
- Similarly for minimising $\mathbf{x}^T\mathbf{M}\mathbf{x}$ with respect to $\mathbf{x}$.

Singular Value Decomposition (SVD)

- Any rectangular m x n matrix **A** with real values can be decomposed as $\mathbf{A}_{mn} = \mathbf{U}_{mm}\mathbf{S}_{mn}\mathbf{V}_{nn}^{T}$ where
  - **U** is an m x m orthogonal matrix ($\mathbf{U}^T\mathbf{U}=\mathbf{I}_m$)
  - **V** is an n x n orthogonal matrix ($\mathbf{V}^T\mathbf{V}=\mathbf{I}_n$) and
  - **S** is an m x n diagonal matrix
- Columns of **U** are orthonormal eigenvectors of $\mathbf{A}\mathbf{A}^T$
- Columns of **V** are orthonormal eigenvectors of $\mathbf{A}^T\mathbf{A}$
- Diagonal of **S** contains the square roots of eigenvalues from **U** or **V** in descending order
  - $\sigma_1 \geq \sigma_2 \geq \ldots \sigma_n$
  - Also called the <u>singular values of A</u>