IM060 Writing and Typesetting with Math

Part 1

Functions, operators, sets, sequences, vectors and matrices

Wilhelm Burger

Interactive Media, Hagenberg

WS 2016/17

Applying functions

Functions are very **powerful concept** in mathematics and well represented in programming (no coincidence).

- A function is **applied** or **evaluated**, e.g., by

$$y = \sin(\alpha).$$

- The **single argument** α is passed to the function named '$\sin$'. α may be any real value, i.e., $\alpha \in \mathbb{R}$, which is the **domain** of this function.
- The function **returns** a single value in $[-1, 1] \subset \mathbb{R}$, which is the **range** (or *image*) of the function.
- The returned value is **assigned** to the variable y.

Functions as mappings: *domain* and *range*

- The fact that the $\sin$ function maps from real numbers to real numbers can be expressed as

$$\sin\colon \mathbb{R} \to \mathbb{R}.$$

- In general, if a *set* A is the **domain** and *set* B is the **range** of a function f,

$$f\colon A \to B.$$

Multiple arguments and return values

- Functions may take **more than one argument**, e.g.,

  $$z = \delta(x, y)$$

  with $x, y \in \mathbb{R}$, i.e., $\delta\colon \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ or $\delta\colon \mathbb{R}^2 \to \mathbb{R}$.
- A function may also **return more than one value**, for example, the "greatest common divisor" function

  $$(d, r) = \gcd(a, b)$$

  with $a, b \in \mathbb{Z}$, returns both the divisor d and the remainder r of the integer division a/b.
- Note that some **programming languages** (unfortunately not the most popular ones) support multiple return values.

Inversion, squaring, composition etc.

- **Inverse of a function:** The expression

$$x = f^{-1}(y)$$

  stands for the value x, such that $y = f(x)$.
- There may be **no such such value** (f is not invertible) or **multiple values**, e.g.,

$$x = \sin^{-1}(0.5)$$

  holds for infinitely many values x.

Inversion, squaring, composition etc.

- **Squaring (exposing) a function:** These expressions are often used synonymously:

$$f^n(x) \equiv [f(x)]^n$$

- Often used with **trigonometric** functions, such as $\sin^2(x) \equiv [\sin(x)]^2$.
- Note that this is **not quite consistent**, because $f^{-1}(x)$ does not mean $\frac{1}{f(x)}$! Thus, use only when the meaning is clear!

Inversion, squaring, composition etc.

- **Composition of functions:** Given two functions $f\colon A \to B$ and $g\colon B \to C$,

$$\boxed{(g \circ f)(x) \equiv g(f(x)),}$$

  denotes the **composition** or **concatenation**[1] of f and g.
- Concatenation defines a **new function**

$$(g \circ f)\colon A \to C.$$

- In general,

$$(f_n \circ \ldots \circ f_2 \circ f_1)(x) \equiv f_n(\ldots f_2(f_1(x)) \ldots).$$

[1] Hintereinanderausführung ($\circ$ = `\circ`)

Standard functions and operators

Standard functions and operators are **written in upright** (roman) font:

- $\sin, \cos, \tan, \arcsin, \arccos, \arctan, \ldots$
  `\sin, \cos, \tan, \arcsin, \arccos, \arctan`
- $\min, \max, \arg, \lim, \ldots$
  `\min, \max, \arg, \lim`
- $\log, \ln, \exp, \gcd, \ldots$
  `\log, \ln, \exp, \gcd`
- $\det, \dim, \bmod$
  `\det, \dim, \bmod`
- . . . (many more)

Always **use the predefined LaTeX commands**!

Defining functions

- A typical function definition:

$$f(x,y) = 3x^2 - 2y + 7xy, \qquad \text{for } 0 \leq x, y < 1$$

- Often, functions are not defined by a **single expression** (as above) but **piecewise** (for different argument values), e.g.,

$$f(x) = \begin{cases} e^{-1/x^2} & \text{for $x > 0$,} \\ 0 & \text{otherwise.} \end{cases}$$

Here is how it is done (w. AMSmath extensions):

```
    f(x) = \begin{cases}
              e^{-1/x^2}  & \text{for $x > 0$,} \\
              0           & \text{otherwise.}
           \end{cases}
```

Text inside math

- Inside any math element, **switch to text mode** with `\text{...}`.
- Inside `\text{...}`, switch **back to math mode** with `$...$`.
- Example:

$$\Phi_{\mathrm{col}}(u,v) = \tan^{-1}\Bigl(\frac{I_{m,y}(u,v)}{I_{m,x}(u,v)}\Bigr), \quad \text{with } m = \max_j E_j(u,v)$$

```
\begin{align}
  \Phi_{\mathrm{col}}(u,v)  &=
  \tan^{-1} \Bigl( \frac{I_{m,y}(u,v)}{I_{m,x}(u,v)} \Bigr),
     \quad \text{with $m = \max_j E_j(u,v)$}
\end{align}
```

- Very convenient, no need to worry about **proper spacing**!

Math inside text (inline math)

- Inline math elements should **fit into the text line** (without additional inter-line spacing). For example, fractions, expressions with exponents, matrices, such as $\dfrac{1}{N}$ or $\displaystyle\sum_{i=0}^{N-1} x_i$ are often **too tall**.
- For **fractions** use $1/N$ or $\tfrac{1}{N}$ (`\tfrac{1}{N}`).
- Inline **sums** are automatically set with **limits on the side**, e.g., $\sum_{i=0}^{N-1} x_i$ (`\sum_{i=0}^{N-1} x_i`).

Derivatives of functions

- **Simple derivatives**: $f', f'', f''', f^{(4)}, \ldots$
  `f', f'', f''', f^{(4)}, \ldots`
- The **partial derivatives** of a function of several variables (x, y) are the derivatives with respect to one of those variables, with the others held constant. E.g., if

$$f(x, y) = x^2 + xy + y^2,$$

the **2** partial derivatives of f (w.r.t. x and y) are

$$\frac{\partial f}{\partial x} = 2x + y \qquad \text{and} \qquad \frac{\partial f}{\partial y} = x + 2y.$$

$\partial =$ `\partial`

What does "=" mean?

The $=$ sign has various different meanings, e.g.:

- $f(x) = x^2 + 1$ … function definition
- $10x + 3y - 7 = 0$ … "is the same" in a typical **equation**,
- if $a = 0$ … equal-operator in a logical expression
- $\pi = 3.14159265359$ … definition of a constant
- $b = b + 1$ … variable assignment (set the value of b to …)

Alternatives to "$=$"

Sometimes it is useful – for **improved clarity** – to replace $=$ by something more specific:

- $f(x) := x^2 + 1$ or $g(x) \triangleq \sqrt{x-1}$ (function definitions)
- $a \equiv 0$, $a \approx 0$, $a \approxeq 0 \dots$ (comparison)
- $b \leftarrow b + 1$ (variable assignment, change of value)

Never use programming operators instead of math operators, e.g.,

- if $a == b \dots$ wrong, use $a = b$ (`a = b`) or $a \equiv b$ (`a \equiv b`).
- if $a \mathrel{!=} b \dots$ wrong, use $a \neq b$ (`a \neq b`).
- if $a <= b \dots$ wrong, use $a \leq b$ (`a \leq b`).
- if $a >= b \dots$ wrong, use $a \geq b$ (`a \geq b`).
- if $a{+}{+} \dots$ wrong, use $a = a + 1$ or $a \leftarrow a + 1$.

Working with sums

- Summation is a **very frequent** operation, e.g.,

$$\frac{1}{N} \cdot \sum_{i=0}^{N-1} (x_i - \bar{x})^2 \qquad \text{or} \qquad \sum_{\mathbf{x} \in C} \|\mathbf{x}\|^2$$

```
\frac{1}{N} \cdot \sum_{i=0}^{N-1} (x_i - \bar{x})^2
\sum_{\mathbf{x} \in C} \|\mathbf{x}\|^2
```

- Sometimes we need **multi-line limits** (spacing we'll handle later):

$$\sum_{\substack{0 \leq i \leq m \\ 0 < j < n}} P(i,j)$$

```
\sum_{\substack{0 \leq i \leq m \\ 0 < j < n}} P(i,j)
```

Sets

- Sets are a **powerful and universal concept** in mathematics. Sets are **not ordered** and contain **no duplicate** elements!
- Typical **set notation**: $A = \{a, b, c\}$

```
A = \{ a, b, c \}
```

- **Empty set**: $A = \{\}$ or $A = \varnothing$

```
A = \{ \}
A = \varnothing
```

- Standard **number sets**: $\mathbb{R}, \mathbb{C}, \mathbb{N}, \mathbb{Z}, \mathbb{Q}$

```
\mathbb{R}, \mathbb{C}, \mathbb{N}, \mathbb{Z}, \mathbb{Q}
```

- **Set generation** (example):
  $B = \{i \in \mathbb{Z} \mid \underbrace{(i \bmod 2) = 0, i > K}_{\text{some logical expression on } i}\}$

Sets

Size of a set:

- $|A|$ or $\mathrm{card}(A)$ denote the size (number of elements, *cardinality*) of the set A.

Testing/comparing sets:

- Testing if an **element** x is (is not) **contained** in set A:
  if $x \in A$, $x \notin A$ ...(`x \in A`, `x \notin A`)
- Testing if A is a **subset** or **superset** of B:
  if $A \subseteq B$, $A \supseteq B$ ...(`A \subseteq B`, `A \supseteq B`)
- Testing if A is a **proper subset** or **superset** of B (i.e., $A \neq B$):
  if $A \subset B$, $A \supset B$ ...(`A \subset B`, `A \supset B`)

Sets

Operations on multiple sets:

- **Union** of two sets A and B:
  $C = A \cup B$ ...(`C = A \cup B`)
- **Intersection** of two sets A and B:
  $C = A \cap B$ ...(`C = A \cap B`)
- **Union** of N sets $A_1, \ldots, A_N$:
  $C = \bigcup_{i=1}^{N} A_i$ ...(`C = \bigcup_{i=1}^{N} A_i`)
- **Intersection** of N sets $A_1, \ldots, A_N$:
  $C = \bigcap_{i=1}^{N} A_i$ ...(`C = \bigcap_{i=1}^{N} A_i`)

Ordered sequences

- A **sequence** (list) contains elements in a particular **ordering**.
- It may also (unlike a set) contain the **same element more than once**.
- Typical sequence notation:
  $S = (a, b, c)$ ... `S = (a, b, c)`
- When the list is **named by a letter**, its elements are typically named analogously:
  $\boldsymbol{a} = (a_1, \ldots, a_N)$ ... `\boldsymbol{a} = (a_1,\ldots,a_N)`
- A **particular element** in a list can be accessed by treating the sequence as a discrete function, e.g.,
  $x = a_i$ or $x = \boldsymbol{a}(i)$
  This can also be used to replace a sequence element, e.g.,
  $\boldsymbol{a}(i) \leftarrow x$.

Vectors

- A vector is simply a **sequence** with **elements of the same type** (typically numbers).
- This is a **row vector** with 3 elements (note the commas!):

$$\boldsymbol{x} = (4, -2, 7)$$

- This is a **column vector** with 2 elements:

$$\boldsymbol{y} = \begin{pmatrix} 1 \\ 6 \end{pmatrix} = (1, 6)^{\mathsf{T}}$$

```
\boldsymbol{y} =
  \begin{pmatrix}
      1 \\
      6
  \end{pmatrix}
  = (1, 6)^{\mathsf{T}}
```

Vector indexes

- **Traditionally**, vector indexes start with **1**, e.g., for a vector of length N,

$$\boldsymbol{x} = (x_1, x_2, \ldots, x_N).$$

- In **programming**, vectors are typically implemented as **arrays**, where (in most modern programming languages) indexes are **0**-based, i.e.,

$$\boldsymbol{x} = (x_0, x_1, \ldots, x_{N-1}).$$

- Using the same indexing throughout your work **helps to avoid confusion**! Decide early and be **consistent**!

More on vectors

- **Transpose:** $\boldsymbol{x}^{\mathsf{T}}$ denotes the **transpose** of the vector $\boldsymbol{x}$. If $\boldsymbol{x}$ is a **column vector** (default), then $\boldsymbol{x}^{\mathsf{T}}$ is a **row vector** (and vice versa):

```
\boldsymbol{x}^{\mathsf{T}}  % many other versions in use
```

- **Zero (column) vector**: $\mathbf{0} = (0, 0, \ldots, 0)^{\mathsf{T}}$, e.g.,

$$\mathbf{A} \cdot \boldsymbol{x} - \boldsymbol{b} = \mathbf{0}$$

- **Norm:** $\|\boldsymbol{x}\|$ commonly denotes the **Euclidean** or **L2** norm of $\boldsymbol{x}$:

```
\| \boldsymbol{x} \|
```

$$\|\boldsymbol{x}\| = \|\boldsymbol{x}\|_2 = \left(\sum_{i=0}^{N-1} x_i^2 \right)^{1/2}, \qquad \|\boldsymbol{x}\|_n = \left(\sum_{i=0}^{N-1} x_i^n \right)^{1/n}$$

Matrices

- A 2×3 matrix, with 3 **rows** $(i = 0, 1, 2)$ and 2 **columns** $(j = 0, 1)$:

$$\mathbf{A} = \begin{pmatrix} A_{0,0} & A_{0,1} \\ A_{1,0} & A_{1,1} \\ A_{2,0} & A_{2,1} \end{pmatrix} \quad \text{or} \quad \mathbf{A} = \begin{bmatrix} A_{0,0} & A_{0,1} \\ A_{1,0} & A_{1,1} \\ A_{2,0} & A_{2,1} \end{bmatrix}$$

```
\mathbf{A} =
  \begin{pmatrix}   % or bmatrix for [..]
     A_{0,0} & A_{0,1} \\
     A_{1,0} & A_{1,1} \\
     A_{2,0} & A_{2,1}
  \end{pmatrix}
```

- Sometimes the **short notation** $\mathbf{A} = (A_{i,j})$ or $\mathbf{A} = [A_{i,j}]$ is used.
- **Identity matrix**: e.g., $\mathbf{I}_2 = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right)$.

Homework assignment

Option A: Optimal Shipping

- Assume that you have a number of products that must be shipped to the same distant location.
- Each item has a particular weight (its only relevant property).
- For shipping, a fleet of trucks is available, each with (a) a specific weight capacity and (b) different (but fixed) trip costs.
- The goal is to minimize the overall costs of shipping, by optimizing the use of each truck and the number of trips.

Formulate this optimization problem in mathematical terms (you don't have to solve it). Use sequences (vectors) in your notation. Wrap your presentation in proper prose.

Homework assignment

Option B: Regular Sampling along a Polygon Path

- Given is a closed 2D polygon (i.e., a sequence of 2D points).
- Starting from a particular vertex, the task is to sample the path at a predefined number of equally spaced points.
- The result should be the sequence of sample points. Describe how to calculate their positions.

Cast this problem into a formal description. Specify all relevant entities, use sequences (vectors) in your description. Wrap your presentation in proper prose.