IM060 Writing and Typesetting with Math

Part 3

Fine-tuning, algorithms, inserting graphics, defining LaTeX commands

Wilhelm Burger

Interactive Media, Hagenberg

WS 2016/17

Parentheses, brackets and other delimiters

- **Standard** sizes: $(\dots), [\dots], \{\dots\}, \langle\dots\rangle$ etc.
  `(...), [...], \{...\}, \langle...\rangle`
- **Automatic** size adjustment: $\left(\dots\right), \left[\dots\right], \left\{\dots\right\}$
  `\left(...\right), \left[...\right], \left\{...\right\}`
- **Manual** size adjustment: $\bigl(\dots\bigr), \Bigl(\dots\Bigr), \biggl(\dots\biggr), \Biggl(\dots\Biggr)$
  `\bigl(...\bigr), \Bigl(...\Bigr), \biggl(...\biggr), \Biggl(...\Biggr)`

More align – \intertext

Alignment across intermediate text – example:

Like its constituting circles, this ellipse is centered at $(x_c, y_c) = (0, 0)$ and performs m revolutions for one traversal of the contour. G_{-m} specifies the circle

$$z_{-m}(\varphi) = G_{-m} \cdot e^{\mathrm{i}(-\varphi)} = r_{-m} \cdot e^{\mathrm{i}(\theta_{-m} - \varphi)}, \tag{7}$$

for $\varphi \in [0, 2\pi]$, with starting angle θ_{-m} and radius r_{-m}, rotating in clockwise direction. Similarly, G_{+m} specifies the circle

$$z_{+m}(\varphi) = G_{+m} \cdot e^{\mathrm{i}(\varphi)} = r_{+m} \cdot e^{\mathrm{i}(\theta_{+m} + \varphi)}, \tag{8}$$

Note the alignment at the $=$ sign across the intermediate paragraph!

More align – \intertext

LaTeX source for the previous example:

```
...
Like its constituting circles, this ellipse is centered at
$(x_c, y_c) = (0,0)$ and performs $m$ revolutions for one
traversal of the contour. $G_{-m}$ specifies the circle
%
\begin{align}
   z_{-m}(\varphi) &= G_{-m} \cdot e^{\I (-\varphi)} =
      r_{-m} \cdot e^{\I (\theta_{-m} - \varphi)} ,
% ---------------------------------------------------------
\intertext{for $\varphi \in [0, 2\pi]$, with starting angle
$\theta_{-m}$ and radius $r_{-m}$, rotating in clockwise
direction. Similarly, $G_{+m}$ specifies the circle}
% ---------------------------------------------------------
   z_{+m}(\varphi) &= G_{+m} \cdot e^{\I (\varphi)} =
      r_{+m} \cdot e^{\I (\theta_{+m} + \varphi)} ,
\end{align}
...
```

split is similar to align (source)

Practical for alignment within a **single multi-line equation**:

$$\begin{split}E_2(u,v) = \|\boldsymbol{E}(u,v)\|_2 &= \left[E_{\mathrm{R}}^2(u,v) + E_{\mathrm{G}}^2(u,v) + E_{\mathrm{B}}^2(u,v)\right]^{1/2} \\ &= \left[I_{\mathrm{R},x}^2 + I_{\mathrm{R},y}^2 + I_{\mathrm{G},x}^2 + I_{\mathrm{G},y}^2 + I_{\mathrm{B},x}^2 + I_{\mathrm{B},y}^2\right]^{1/2}\end{split} \tag{9}$$

Note that Eq. 9 has two lines but only a *single* label!

`split` is similar to `align`

LaTeX source for the previous example:

```
\begin{equation}
   \begin{split}
      E_{2}(u,v) &= \left\| \boldsymbol{E}(u,v) \right\|_2
        = \left[ E_{\mathrm{R}}^2(u,v) + E_{\mathrm{G}}^2(u,v)
        + E_{\mathrm{B}}^2(u,v) \right]^{1/2}
   \\
      &= \left[
         I_{\mathrm{R},x}^2  + I_{\mathrm{R},y}^2  +
         I_{\mathrm{G},x}^2  + I_{\mathrm{G},y}^2  +
         I_{\mathrm{B},x}^2  + I_{\mathrm{B},y}^2
         \right]^{1/2}
   \end{split}
\label{eq:EdgeMagnitude}
\end{equation}
```

alignat – aligning *multiple* positions

In this case, the equations in Eqn. 7.15 simplify to

$$\begin{alignat}{4} y_1 &= a && - b + {} && c , \tag{10}\\ y_2 &= && && c , \tag{11}\\ y_3 &= a && + b + {} && c , \tag{12} \end{alignat}$$

```
In this case, the equations in Eqn.~\ref{xxx}
simplify to
\begin{alignat}{4}
   y_1 &=  a  & & - b + {} & & c , \\
   y_2 &=     & &           & & c , \\
   y_3 &=  a  & & + b + {} & & c ,
\end{alignat}
```

Note the + {} to give + operator a second operand. Try without!

Underbraces

Underbraces are useful for structuring/explaining large expressions.

$$I'(u,v) = \frac{1}{W_{u,v}} \cdot \sum_{\substack{i= \\ -\infty}}^{\infty} \sum_{\substack{j= \\ -\infty}}^{\infty} I(i,j) \cdot \underbrace{H_{\mathrm{d}}(i-u, j-v) \cdot H_{\mathrm{r}}\big(I(i,j) - I(u,v)\big)}_{w_{i,j}}$$

`\underbrace{expression}_{subtext}`

\displaystyle

`\displaystyle` can be used to switch (revert) to the font size / positioning style used for “displayed” math:

- `x = a_0 + \frac{1}{a_1 + \frac{1}{a_2}}`

$$x = a_0 + \frac{1}{a_1 + \frac{1}{a_2}}$$

- `x = a_0 + \frac{1}{a_1 + \displaystyle\frac{1}{a_2}}`

$$x = a_0 + \frac{1}{a_1 + \displaystyle\frac{1}{a_2}}$$

\textstyle

`\textstyle` can be used to switch (revert) to the font size / positioning style that is used for “in-line” math:

- `m = \sum_{i=0}^{N-1} i^2`

$$m = \sum_{i=0}^{N-1} i^2$$

- `m = \textstyle\sum_{i=0}^{N-1} i^2`

$$m = \textstyle\sum_{i=0}^{N-1} i^2$$

Horizontal spacing in math mode

- Predefined **positive** spacing commands:

full:	*short:*
`\thinspace`	`\,`
`\medspace`	`\:`
`\thickspace`	`\;`
`\quad`	
`\qquad`	

Example (note the spacing before the final dot):

$$\bar{y} = \frac{1}{N} \cdot \sum_{i=1}^{N} v_i. \qquad \text{vs.} \qquad \bar{y} = \frac{1}{N} \cdot \sum_{i=1}^{N} v_i \; .$$

```
   \bar{y} = \frac{1}{N} \cdot \sum_{i = 1}^{N} v_i \; .
```

Horizontal spacing in math mode

- Predefined **negative** spacing commands:

full:	*short:*
`\negthinspace`	`\!`
`\negmedspace`	
`\negthickspace`	

$$\bar{y} = \frac{1}{N} \cdot \sum_{0\leq i<N} v_i \qquad \text{vs.} \qquad \bar{y} = \frac{1}{N} \cdot \!\!\! \sum_{0\leq i<N} \!\!\! v_i$$

```
   .. = \frac{1}{N} \cdot \!\!\! \sum_{0 \leq i < N} \!\!\! v_i
```

Note: We shall see more elegant methods for fixing this!

Horizontal spacing in math mode

General math spacing commands (use in math mode only):

- `\mspace{24mu}`
- `\mspace{-18mu}`
- `mu` = "math unit" ($1\,\mathrm{mu} = 1/18\,\mathrm{em}$)

$$\bar{y} = \frac{1}{N} \cdot \sum_{0 \leq i < N} v_i \qquad \text{vs.} \qquad \bar{y} = \frac{1}{N} \cdot \mspace{-12mu} \sum_{0 \leq i < N} \mspace{-12mu} v_i$$

```
.. \cdot \mspace{-12mu} \sum_{0 \leq i < N} \mspace{-12mu} v_i
```

`\phantom`

With `\phantom` you can insert **invisible elements** for exact spacing:

$$\boldsymbol{A} = \begin{pmatrix} 3 & -2 \\ -4 & 1 \end{pmatrix} \qquad \text{vs.} \qquad \boldsymbol{A} = \begin{pmatrix} 3 & -2 \\ -4 & 1 \end{pmatrix} \tag{13}$$

```
boldsymbol{A} =
      \begin{pmatrix}
            \phantom{-}3 & -2 \\
            -4 & \phantom{-}1
      \end{pmatrix}
```

Also see `\vphantom`.

smash-ing vertical length

Sometimes we want to insert **elements that have no height**:

- `\smash{..}` zeros the **vertical** height of its argument.
- Works in **text mode** and **math mode**.
- Useful to **avoid extra line spacing** from large math constructs.
- Compare boxes for $\boxed{X^{2^4}}$ and $\underline{X^{2^4}}$
  ```
  Compare boxes for $X^{2^4}$ and $\smash{X^{2^4}}$
  ```
- Example: $\sqrt{\sigma_{\mathrm{init}}^2 - \sigma_{\mathrm{s}}^2}$ vs. $\sqrt{\sigma_{\smash{\mathrm{init}}}^2 - \sigma_{\mathrm{s}}^2}$

```
\sqrt{\sigma_{\smash{\mathrm{init}}}^2 ... }
```

\clap and friends (*smashing* horizontal length)

- **Text mode:** `\llap{..}`, `\clap{..}`, `\rlap{..}` eliminate the **horizontal** length of their arguments (standard LaTeX).
- **Math mode:** `mathtools` package provides `\mathllap{..}`, `\mathclap{..}`, `\mathrlap{..}`

Default: `\sum_{(i,j) \in \mathcal{R}_1} = ...`

$$\bar{\boldsymbol{c}} = \sum_{(i,j)\in\mathcal{R}_1} \boldsymbol{c}_1 \cdot H(i,j) + \sum_{(i,j)\in\mathcal{R}_2} \boldsymbol{c}_2 \cdot H(i,j) \tag{14}$$

Better: `\sum_{\mathclap{(i,j) \in \mathcal{R}_1}} = ...`

$$\bar{\boldsymbol{c}} = \sum_{\mathclap{(i,j)\in\mathcal{R}_1}} \boldsymbol{c}_1 \cdot H(i,j) + \sum_{\mathclap{(i,j)\in\mathcal{R}_2}} \boldsymbol{c}_2 \cdot H(i,j) \tag{15}$$

Adjusting vertical spacing in multi-line equations

- The new-line command \\ accepts an optional length parameter: `\\[len]`, which adds `len` vertical space (may be negative):

$$\begin{split}
\mathsf{h}_{\nabla}(i_0,j_0,k_0) & \leftarrow z \cdot \alpha_0 \cdot \beta_0 \cdot \gamma_0,\\
\mathsf{h}_{\nabla}(i_1,j_0,k_0) & \leftarrow z \cdot \alpha_1 \cdot \beta_0 \cdot \gamma_0,\\[-6pt]
\vdots\qquad & \qquad\qquad\vdots \\[-4pt]
\mathsf{h}_{\nabla}(i_1,j_1,k_1) & \leftarrow z \cdot \alpha_1 \cdot \beta_1 \cdot \gamma_1.
\end{split} \tag{16}$$

```
\begin{equation}
\begin{split}
\mathsf{h}_{\nabla}(i_0,j_0,k_0) &
   \leftarrow z \cdot \alpha_0 \cdot \beta_0 \cdot \gamma_0,\\
\mathsf{h}_{\nabla}(i_1,j_0,k_0) &
   \leftarrow z \cdot \alpha_1 \cdot \beta_0 \cdot \gamma_0,\\[-6pt]
\vdots\qquad & \qquad\qquad\vdots \\[-4pt]
\mathsf{h}_{\nabla}(i_1,j_1,k_1) &
   \leftarrow z \cdot \alpha_1 \cdot \beta_1 \cdot \gamma_1.
\end{split}
\end{equation}
```

Footnotes in equations?

- Footnote marks may be confused with exponents.
- Do not work properly.
- Not a good idea, **simply avoid**!

Defining new LaTeX commands

Create your own commands for less work, to make your texts more readable and to allow easy replacement later on.

- Commands without an argument:
  `\newcommand{\R}{\mathbb{R}}` … `$\R$` → $\mathbb{R}$
  `\newcommand{\True}{\mathsf{true}}` … `$\True$` → true
- Commands with one argument:
  `\newcommand{\eqnref}[1]{Eqn.~\ref{#1}}`
  `\eqnref{foo}` → Eqn. 15
  `\newcommand{\Map}[1]{\ensuremath{\mathsf{#1}}}`
  `$\Map{H} \colon [1,N] \mapsto \R$`
  $\mathsf{H}\colon [1,N] \mapsto \mathbb{R}$

Defining math operators

- `\DeclareMathOperator{\argmin}{argmin}`
  `\argmin_{\alpha} f(\alpha)` ... → $\operatorname{argmin}_\alpha f(\alpha)$
- `\DeclareMathOperator*{\argmax}{argmax}`
  `\argmax_{\beta} f(\beta)` ... → $\operatorname*{argmax}_\beta f(\beta)$
- Alternatively use `\operatorname{..}` when needed:
  $0011_b \operatorname{and} 1010_b = 0010_b$

```
    $0011_b \operatorname{and} 1010_b = 0010_b$
```

Page breaks in mathematical blocks (1)

- **Globally** allow page breaks in multi-line equations by
  `\allowdisplaybreaks[n]`
  `amsldoc.pdf`: *An optional argument $n = 1, \ldots, 4$ can be used for finer control: 1 means allow page breaks, but avoid them as much as possible; values of 2, 3, 4 mean increasing permissiveness.*
- **Locally** (inside an equation) use `\displaybreak[n]`:

$$\sigma^2_{\mathrm{w}}(q) = \mathsf{P}_0(q) \cdot \sigma^2_0(q) + \mathsf{P}_1(q) \cdot \sigma^2_1(q) \tag{17}$$

$$= \frac{1}{N} \bigl[n_0(q) \cdot \sigma^2_0(q) + n_1(q) \cdot \sigma^2_1(q) \bigr] \tag{18}$$

```
\begin{align}
\sigma^2_{\mathrm{w}} (q)
  &= \mathsf{P}_0(q) \cdot \sigma^2_0(q)
     + \mathsf{P}_1(q) \cdot \sigma^2_1(q)
\displaybreak[4]\\  % <-- strongly encourage a page break here
  &= \frac{1}{N} \bigl[ n_0(q) \cdot \sigma^2_0(q)
     + n_1(q) \cdot \sigma^2_1(q) \bigr]
\end{align}
```

Page breaks in mathematical blocks (2)

- **Note:** Certain equation environments wrap their contents in an **unbreakable box**, with the consequence that **neither** `\displaybreak` **nor** `\allowdisplaybreaks` have any effect on them:

  These include `split`, `aligned`, `gathered`, and `alignedat`.

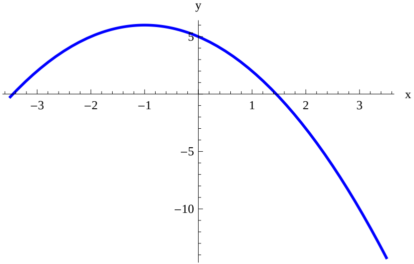
Line breaks in inline-math

Sometimes **inline math** elements such as $a + b + c + d + e + f + g + h + i + j + k + l$ become long and create odd line breaks.

- **Fix:** Insert `\allowbreak` at a suitable position:
- Sometimes inline math elements such as $a + b + c + d + e + f + g + h + i + j + k + l$ become long and create odd line break.
  `$a + b + c + d + e + \allowbreak f + g + h + i + j + k + l$`
- It may be necessary to **split** math elements **into multiple pieces**.
- To **avoid line breaks**, use additional `{..}` brackets, e.g., `${f(x) = x^2 + 2}$` will never break at the $=$ sign.

Inserting graphics (1)

A simple plot created with Mathematica (`PlotA` in file `plots.nb`):



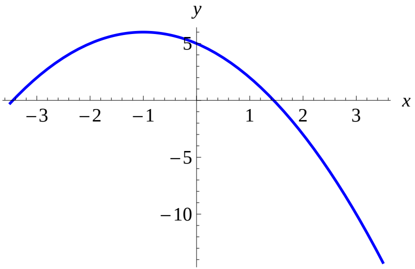
Mathematica:

```
PlotA = Plot[-x^2 - 2 x + 5, {x, -3.5, 3.5},
        PlotStyle -> {Blue, Thick}, AxesLabel -> {"x", "y"}]
Export["plotA.pdf", PlotA]
```

```
    \begin{center}
      \includegraphics[width=0.5\textwidth]{mathematica/plotA}
    \end{center}
```

Inserting graphics (2)

Improved version, with font style and size adapted (`PlotB`):

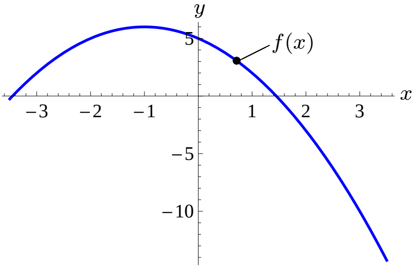


Mathematica:

```
PlotB = Plot[-x^2 - 2 x + 5, {x, -3.5, 3.5}, PlotStyle -> {Blue, Thick},
  BaseStyle -> {FontFamily -> "Times", FontSize -> 14},
  AxesLabel -> {Style["x", Italic], Style["y", Italic]}]
Export["plotB.pdf", PlotB]
```

Inserting graphics (3)

PlotC: labels added in LaTeX (using the overpic package)



```
\begin{center}\footnotesize
   \begin{overpic}[width=0.5\textwidth]{mathematica/plotC}
      \put(97.5,40.5){$x$}       % relative positions (in %)
      \put(47,61.5){$y$}
      \put(57.5,50){\circle*{2}}
      \thinlines\put(58,50){\line(2,1){7.5}}
      \put(66,53){$f(x)$}
   \end{overpic}
\end{center}
```

Inserting graphics – other options

Other options for “scientific” illustrations:

- **psfrag** package: Allows replacement of specific character strings in an EPS graphic file. Only works with traditional DVI/PostScript workflow (obsolete)!
- Use a vector graphics tool (e.g., **Adobe Illustrator**) and TeX-like fonts (e.g., BaKoMa TrueType fonts).
- Use a “LaTeX-aware” drawing program (e.g., **Inkscape**).
- Other LaTeX graphics packages, e.g., **TikZ** and **PGF**.
- Use **vector graphics** whenever possible!
  Never use PhotoShop to draw an illustration!

Homework assignment 3

Create a “scientific”' illustration and include it in your document:

- For the “**Regular Sampling along a Polygon Path**” problem (see assignment 2), make a clean illustration that describes the situation and the important entities.
- Use a ***vector* drawing tool** (*Adobe Illustrator*, PowerPoint, *Inkscape*[4], *LibreOffice-Draw*[5], ...) and export/save the result as a **PDF** (crop with Acrobat if necessary).
- Include the resulting PDF in your LaTeX document and place it inside a `figure` environment, including a caption text (see the thesis template[6] for examples).
- Use `overpic` (see previous slides) to place mathematical LaTeX symbols on top of your illustration.

[4] https://inkscape.org/

[5] https://www.libreoffice.org/

[6] https://github.com/Digital-Media/HagenbergThesis

The End

Related topics **you might want to explore**:

- Designing and typesetting algorithms (see Hagenberg thesis template).
- Creating technical/scientific graphics and illustrations.
- Presentations (slides) made with LaTeX (tedious).
- . . . ?