For office use only	Team Control Number	For office use only
T1 ________	**8889888**	F1 ________
T2 ________		F2 ________
T3 ________	Problem Chosen	F3 ________
T4 ________	**B**	F4 ________

The 2019 Mathematical Contest in Modeling (MCM) Summary Sheet

My Title

Summary

This paper showes the usage of latex class HZNUMCM and gives most used examples for mathematical typeset. This paper can also be used as a handbook for typing latex. We give the pdf file and the source code file.

This HZNUMCM class file comes from easymcm, a latex .sty package and a simple MCM Contest Report Template for basic users from oringinal template by latexstudio, <latexstudio@hotmail.com> improved by youjiarui189 (xjtu-blacksmith), <yjr134@163.com>.

This document also pick many pages from the not so not introduction to Latex [1], so that one can search quickly.

Contents

1 Basic usage of the HZNUMCM class

1.1 Basic document structure for using HZNUMCM class

The basic .tex file structure is as the following and comment for each corespending line is listed.

```
\documentclass{HZNUMCM}.    % class choose
\setControlNumber{8889888}  % set Control Number
\setContestType{MCM}        % contest type, only MCM or ICM allowed
\setProblemLetter{B}        % choose problem A,B,C,D,E or F
\setBibFilename{McmTexExample} % set bib file, without dot and suffix
\setPaperTitle{My Title}    % set your title
\setSummary{summary}        % define summary
\begin{document}
   \showSummarySheet       % show summary, the first page.
   \showContents           % show contents
   \section{aaaa}
      ...
   \section{bbbb}
      ...
      ...
   \showReferences         % show references
\end{document}
```

1.2 Process from source files to target .pdf

Usually, the process can be executed as following:

- Type what you think in the .tex file, following the Latex syntax.
- xelatex several times, biblatex one time and then xelatex several times again.

2 Introduction to mathematical typeseting within Latex

We choose mathematical typeseting from [1].

3 Some complements

We put the source code of the HZNUMCM latex class file and this file on the gitHub `https://github.com/hznu-maguochun/HZNUMCM`, so that one use it can understand it. Any one, who want to use or improve it, is welcome. This class file also gives an idea about how to make a formatted file.

References

[1] Irene Hyna Tobias Oetiker, Hubert Partl and Elisabeth Schlegl. The not so short introduction to latex2e. `CTAN://info/lshort`. Online; accessed 19 January 2019.

Chapter 3

Typesetting Mathematical Formulae

Now you are ready! In this chapter, we will attack the main strength of TEX: mathematical typesetting. But be warned, this chapter only scratches the surface. While the things explained here are sufficient for many people, don't despair if you can't find a solution to your mathematical typesetting needs here. It is highly likely that your problem is addressed in AMS-LATEX.

3.1 The AMS-LATEX bundle

If you want to typeset (advanced) mathematics, you should use AMS-LATEX. The AMS-LATEX bundle is a collection of packages and classes for mathematical typesetting. We will mostly deal with the amsmath package which is a part of the bundle. AMS-LATEX is produced by The *American Mathematical Society* and it is used extensively for mathematical typesetting. LATEX itself does provide some basic features and environments for mathematics, but they are limited (or maybe it's the other way around: AMS-LATEX is *unlimited*!) and in some cases inconsistent.

AMS-LATEX is a part of the required distribution and is provided with all recent LATEX distributions.[1] In this chapter, we assume amsmath is loaded in the preamble; `\usepackage{amsmath}`.

3.2 Single Equations

A mathematical formula can be typeset in-line within a paragraph (*text style*), or the paragraph can be broken and the formula typeset separately (*display style*). Mathematical equations *within* a paragraph are entered between `$` and `$`:

[1] If yours is missing it, go to CTAN://pkg/amslatex.

```
Add $a$ squared and $b$ squared
to get $c$ squared. Or, using
a more mathematical approach:
$a^2 + b^2 = c^2$
```

Add a squared and b squared to get c squared. Or, using a more mathematical approach: $a^2 + b^2 = c^2$

```
\TeX{} is pronounced as
$\tau\epsilon\chi$\\[5pt]
100~m$^{3}$ of water\\[5pt]
This comes from my $\heartsuit$
```

TEX is pronounced as $\tau\epsilon\chi$

100 m^{3} of water

This comes from my $\heartsuit$

If you want your larger equations to be set apart from the rest of the paragraph, it is preferable to *display* them rather than to break the paragraph apart. To do this, you enclose them between `\begin{equation}` and `\end{equation}`.[2] You can then `\label` an equation number and refer to it somewhere else in the text by using the `\eqref` command. If you want to name the equation something specific, you `\tag` it instead.

```
Add $a$ squared and $b$ squared
to get $c$ squared. Or, using
a more mathematical approach
 \begin{equation}
   a^2 + b^2 = c^2
 \end{equation}
Einstein says
 \begin{equation}
   E = mc^2 \label{clever}
 \end{equation}
He didn't say
 \begin{equation}
  1 + 1 = 3 \tag{dumb}
 \end{equation}
This is a reference to
\eqref{clever}.
```

Add a squared and b squared to get c squared. Or, using a more mathematical approach

$$a^2 + b^2 = c^2 \tag{3.1}$$

Einstein says

$$E = mc^2 \tag{3.2}$$

He didn't say

$$1 + 1 = 3 \tag{dumb}$$

This is a reference to (3.2).

If you don't want LaTeX to number the equations, use the starred version of `equation` using an asterisk, `equation*`, or even easier, enclose the equation in `\[` and `\]`:[3]

[2] This is an `amsmath` command. If you don't have access to the package for some obscure reason, you can use LaTeX's own `displaymath` environment instead.

[3] This is again from `amsmath`. Standard LaTeX's has only the `equation` environment without the star.

```
Add $a$ squared and $b$ squared
to get $c$ squared. Or, using
a more mathematical approach
 \begin{equation*}
   a^2 + b^2 = c^2
 \end{equation*}
or you can type less for the
same effect:
 \[ a^2 + b^2 = c^2 \]
```

Add a squared and b squared to get c squared. Or, using a more mathematical approach

$$a^2 + b^2 = c^2$$

or you can type less for the same effect:

$$a^2 + b^2 = c^2$$

While `\[` is short and sweet, it does not allow switching between numbered and not numbered style as easily as `equation` and `equation*`.

Note the difference in typesetting style between text style and display style equations:

```
This is text style:
$\lim_{n \to \infty}
 \sum_{k=1}^n \frac{1}{k^2}
 = \frac{\pi^2}{6}$.
And this is display style:
 \begin{equation}
  \lim_{n \to \infty}
  \sum_{k=1}^n \frac{1}{k^2}
   = \frac{\pi^2}{6}
 \end{equation}
```

This is text style: $\lim_{n\to\infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$.
And this is display style:

$$\lim_{n\to\infty} \sum_{k=1}^{n} \frac{1}{k^2} = \frac{\pi^2}{6} \tag{3.3}$$

In text style, enclose tall or deep math expressions or sub expressions in `\smash`. This makes LaTeX ignore the height of these expressions. This keeps the line spacing even.

```
A $d_{e_{e_p}}$ mathematical
expression  followed by a
$h^{i^{g^h}}$ expression. As
opposed to a smashed
\smash{$d_{e_{e_p}}$} expression
followed by a
\smash{$h^{i^{g^h}}$} expression.
```

A $d_{e_{e_p}}$ mathematical expression followed by a $h^{i^{g^h}}$ expression. As opposed to a smashed $d_{e_{e_p}}$ expression followed by a $h^{i^{g^h}}$ expression.

3.2.1 Math Mode

There are also differences between *math mode* and *text mode*. For example, in *math mode*:

1. Most spaces and line breaks do not have any significance, as all spaces are either derived logically from the mathematical expressions, or have to be specified with special commands such as `\,`, `\quad` or `\qquad` (we'll get back to that later, see section 3.7).

2. Empty lines are not allowed. Only one paragraph per formula.

3. Each letter is considered to be the name of a variable and will be typeset as such. If you want to typeset normal text within a formula (normal upright font and normal spacing) then you have to enter the text using the `\text{...}` command (see also section 3.8 on page 60).

```
$\forall x \in \mathbf{R}:
 \qquad x^{2} \geq 0$
```

$\forall x \in \mathbf{R}: \qquad x^{2} \geq 0$

```
$x^{2} \geq 0\qquad
 \text{for all }x\in\mathbf{R}$
```

$x^{2} \geq 0\qquad \text{for all }x\in\mathbf{R}$

Mathematicians can be very fussy about which symbols are used: it would be conventional here to use the 'blackboard bold' font, which is obtained using `\mathbb` from the package amssymb.[4] The last example becomes

```
$x^{2} \geq 0\qquad
 \text{for all } x
 \in \mathbb{R}$
```

$x^{2} \geq 0\qquad \text{for all } x \in \mathbb{R}$

See Table 3.14 on page 69 and Table 6.4 on page 109 for more math fonts.

3.3 Building Blocks of a Mathematical Formula

In this section, we describe the most important commands used in mathematical typesetting. Most of the commands in this section will not require amsmath (if they do, it will be stated clearly), but load it anyway.

Lowercase Greek letters are entered as `\alpha`, `\beta`, `\gamma`, ..., uppercase letters are entered as `\Gamma`, `\Delta`, ...[5]

Take a look at Table 3.2 on page 65 for a list of Greek letters.

```
$\lambda,\xi,\pi,\theta,
 \mu,\Phi,\Omega,\Delta$
```

$\lambda,\xi,\pi,\theta,\mu,\Phi,\Omega,\Delta$

Exponents, Superscripts and Subscripts can be specified using the `^` and the `_` characters. Most math mode commands act only on the next

[4] amssymb is not a part of the $\mathcal{A}\mathcal{M}\mathcal{S}$-LaTeX bundle, but it is perhaps still a part of your LaTeX distribution. Check your distribution or go to `CTAN:/fonts/amsfonts/latex/` to obtain it.

[5] There is no uppercase Alpha, Beta etc. defined in LaTeX 2$_\varepsilon$ because it looks the same as a normal roman A, B...

character, so if you want a command to affect several characters, you have to group them together using curly braces: `{...}`.

Table 3.3 on page 66 lists a lot of binary relations like $\subseteq$ and $\perp$.

```
$p^3_{ij} \qquad
 m_\text{Knuth}\qquad
\sum_{k=1}^3 k \\[5pt]
 a^x+y \neq a^{x+y}\qquad
 e^{x^2} \neq {e^x}^2$
```

$$p^3_{ij} \qquad m_\text{Knuth} \qquad \textstyle\sum_{k=1}^3 k$$
$$a^x+y \neq a^{x+y} \qquad e^{x^2} \neq {e^x}^2$$

The **square root** is entered as `\sqrt`; the n^{th} root is generated with `\sqrt[`n`]`. The size of the root sign is determined automatically by LaTeX. If just the sign is needed, use `\surd`.

See various kinds of arrows like $\hookrightarrow$ and $\rightleftharpoons$ on Table 3.6 on page 67.

```
$\sqrt{x} \Leftrightarrow x^{1/2}
 \quad \sqrt[3]{2}
 \quad \sqrt{x^{2} + \sqrt{y}}
 \quad \surd[x^2 + y^2]$
```

$$\sqrt{x} \Leftrightarrow x^{1/2} \quad \sqrt[3]{2} \quad \sqrt{x^{2} + \sqrt{y}} \quad \surd[x^2 + y^2]$$

While the **dot** sign to indicate the multiplication operation is normally left out, it is sometimes written to help the eye in grouping a formula. Use `\cdot` to typeset a single centered dot. `\cdots` is three centered **dots** while `\ldots` sets the dots low (on the baseline). Besides that, there are `\vdots` for vertical and `\ddots` for diagonal dots. There are more examples in section 3.6.

```
$\Psi = v_1 \cdot v_2
 \cdot \ldots \qquad
 n! = 1 \cdot 2
 \cdots (n-1) \cdot n$
```

$$\Psi = v_1 \cdot v_2 \cdot \ldots \qquad n! = 1 \cdot 2 \cdots (n-1) \cdot n$$

The commands `\overline` and `\underline` create **horizontal lines** directly over or under an expression:

```
$0.\overline{3} =
 \underline{\underline{1/3}}$
```

$$0.\overline{3} = \underline{\underline{1/3}}$$

The commands `\overbrace` and `\underbrace` create long **horizontal braces** over or under an expression:

```
$\underbrace{\overbrace{a+b+c}^6
 \cdot \overbrace{d+e+f}^7}
 _\text{meaning of life} = 42$
```

$$\underbrace{\overbrace{a+b+c}^6 \cdot \overbrace{d+e+f}^7}_\text{meaning of life} = 42$$

To add mathematical accents such as **small arrows** or **tilde** signs to variables, the commands given in Table 3.1 on page 65 might be useful. Wide hats and tildes covering several characters are generated with `\widetilde` and `\widehat`. Notice the difference between `\hat` and `\widehat` and the placement of `\bar` for a variable with subscript. The apostrophe mark `'` gives a prime:

```
$f(x) = x^2 \qquad f'(x)
 = 2x \qquad f''(x) = 2\\[5pt]
 \hat{XY} \quad \widehat{XY}
 \quad \bar{x_0} \quad \bar{x}_0$
```

$f(x) = x^2 \qquad f'(x) = 2x \qquad f''(x) = 2$

$\hat{XY} \quad \widehat{XY} \quad \bar{x_0} \quad \bar{x}_0$

Vectors are often specified by adding small arrow symbols on the tops of variables. This is done with the `\vec` command. The two commands `\overrightarrow` and `\overleftarrow` are useful to denote the vector from A to B:

```
$\vec{a} \qquad
 \vec{AB} \qquad
 \overrightarrow{AB}$
```

$\vec{a} \qquad \vec{AB} \qquad \overrightarrow{AB}$

Names of functions are often typeset in an upright font, and not in italics as variables are, so LaTeX supplies the following commands to typeset the most common function names:

`\arccos`	`\cos`	`\csc`	`\exp`	`\ker`	`\limsup`
`\arcsin`	`\cosh`	`\deg`	`\gcd`	`\lg`	`\ln`
`\arctan`	`\cot`	`\det`	`\hom`	`\lim`	`\log`
`\arg`	`\coth`	`\dim`	`\inf`	`\liminf`	`\max`
`\sinh`	`\sup`	`\tan`	`\tanh`	`\min`	`\Pr`
`\sec`	`\sin`				

```
\begin{equation*}
  \lim_{x \rightarrow 0}
  \frac{\sin x}{x}=1
\end{equation*}
```

$$\lim_{x \rightarrow 0} \frac{\sin x}{x}=1$$

For functions missing from the list, use the `\DeclareMathOperator` command. There is even a starred version for functions with limits. This command works only in the preamble so the commented lines in the example below must be put into the preamble.

```
%\DeclareMathOperator{\argh}{argh}
%\DeclareMathOperator*{\nut}{Nut}
\begin{equation*}
  3\argh = 2\nut_{x=1}
\end{equation*}
```

$$3 \operatorname{argh} = 2 \operatorname*{Nut}_{x=1}$$

For the modulo function, there are two commands: `\bmod` for the binary operator "$a \bmod b$" and `\pmod` for expressions such as "$x \equiv a \pmod{b}$:"

```
$a\bmod b \\
 x\equiv a \pmod{b}$
```

$a \bmod b$
$x \equiv a \pmod{b}$

A built-up **fraction** is typeset with the `\frac{...}{...}` command. In in-line equations, the fraction is shrunk to fit the line. This style is obtainable in display style with `\tfrac`. The reverse, i.e. display style fraction in text, is made with `\dfrac`. Often the slashed form 1/2 is preferable, because it looks better for small amounts of 'fraction material:'

```
In display style:
\begin{equation*}
  3/8 \qquad \frac{3}{8}
  \qquad \tfrac{3}{8}
\end{equation*}
```

In display style:

$$3/8 \qquad \frac{3}{8} \qquad \tfrac{3}{8}$$

```
In text style:
$1\frac{1}{2}$~hours \qquad
$1\dfrac{1}{2}$~hours
```

In text style: $1\frac{1}{2}$ hours $\qquad$ $1\dfrac{1}{2}$ hours

Here the `\partial` command for partial derivatives is used:

```
\begin{equation*}
  \sqrt{\frac{x^2}{k+1}}\qquad
  x^\frac{2}{k+1}\qquad
  \frac{\partial^2f}
  {\partial x^2}
\end{equation*}
```

$$\sqrt{\frac{x^2}{k+1}}\qquad x^\frac{2}{k+1}\qquad \frac{\partial^2f}{\partial x^2}$$

To typeset binomial coefficients or similar structures, use the command `\binom` from amsmath:

```
Pascal's rule is
\begin{equation*}
 \binom{n}{k} =\binom{n-1}{k}
 + \binom{n-1}{k-1}
\end{equation*}
```

Pascal's rule is

$$\binom{n}{k} =\binom{n-1}{k} + \binom{n-1}{k-1}$$

For binary relations it may be useful to stack symbols over each other. `\stackrel{#1}{#2}` puts the symbol given in #1 in superscript-like size over #2 which is set in its usual position.

```
\begin{equation*}
 f_n(x) \stackrel{*}{\approx} 1
\end{equation*}
```

$$f_n(x) \stackrel{*}{\approx} 1$$

The **integral operator** is generated with `\int`, the **sum operator** with `\sum`, and the **product operator** with `\prod`. The upper and lower limits are specified with `^` and `_` like superscripts and subscripts:

```
\begin{equation*}
\sum_{i=1}^n \qquad
\int_0^{\frac{\pi}{2}} \qquad
\prod_\epsilon
\end{equation*}
```

$$\sum_{i=1}^n \qquad \int_0^{\frac{\pi}{2}} \qquad \prod_\epsilon$$

To get more control over the placement of indices in complex expressions, amsmath provides the `\substack` command:

```
\begin{equation*}
\sum^n_{\substack{0<i<n \\
        j\subseteq i}}
   P(i,j) = Q(i,j)
\end{equation*}
```

$$\sum^n_{\substack{0<i<n \\ j\subseteq i}} P(i,j) = Q(i,j)$$

LaTeX provides all sorts of symbols for **bracketing** and other **delimiters** (e.g. $[\ \langle\ \|\ \updownarrow$). Round and square brackets can be entered with the corresponding keys and curly braces with `\{`, but all other delimiters are generated with special commands (e.g. `\updownarrow`).

```
\begin{equation*}
{a,b,c} \neq \{a,b,c\}
\end{equation*}
```

$$a,b,c \neq \{a,b,c\}$$

If you put `\left` in front of an opening delimiter and `\right` in front of a closing delimiter, LaTeX will automatically determine the correct size of the delimiter. Note that you must close every `\left` with a corresponding `\right`. If you don't want anything on the right, use the invisible "`\right.`":

```
\begin{equation*}
1 + \left(\frac{1}{1-x^{2}}
    \right)^3 \qquad
\left. \ddagger \frac{~}{~}\right)
\end{equation*}
```

$$1 + \left(\frac{1}{1-x^{2}}\right)^3 \qquad \left. \ddagger \frac{~}{~}\right)$$

In some cases it is necessary to specify the correct size of a mathematical delimiter by hand, which can be done using the commands `\big`, `\Big`, `\bigg` and `\Bigg` as prefixes to most delimiter commands:

```
$\Big((x+1)(x-1)\Big)^{2}$\\
$\big( \Big( \bigg( \Bigg( \quad
\big\} \Big\} \bigg\} \Bigg\} \quad
\big\| \Big\| \bigg\| \Bigg\| \quad
\big\Downarrow \Big\Downarrow
\bigg\Downarrow \Bigg\Downarrow$
```

$\Big((x+1)(x-1)\Big)^{2}$

$\big(\Big(\bigg(\Bigg(\quad \big\} \Big\} \bigg\} \Bigg\} \quad \big\| \Big\| \bigg\| \Bigg\| \quad \big\Downarrow \Big\Downarrow \bigg\Downarrow \Bigg\Downarrow$

For a list of all delimiters available, see Table 3.8 on page 68.

3.4 Single Equations that are Too Long: multline

If an equation is too long, we have to wrap it somehow. Unfortunately, wrapped equations are usually less easy to read than not wrapped ones. To improve the readability, there are certain rules on how to do the wrapping:

1. In general one should always wrap an equation **before** an equality sign or an operator.
2. A wrap before an equality sign is preferable to a wrap before any operator.
3. A wrap before a plus- or minus-operator is preferable to a wrap before a multiplication-operator.
4. Any other type of wrap should be avoided if at all possible.

The easiest way to achieve such a wrapping is the use of the `multline` environment:[6]

```
\begin{multline}
  a + b + c + d + e + f
  + g + h + i
  \\
  = j + k + l + m + n
\end{multline}
```

$$
\begin{multline}
a+b+c+d+e+f+g+h+i \\
= j+k+l+m+n \tag{3.4}
\end{multline}
$$

The difference from the `equation` environment is that an arbitrary line-break (or also multiple line-breaks) can be introduced. This is done by putting a \\ on those places where the equation needs to be wrapped. Similarly to `equation*` there also exists a `multline*` version for preventing an equation number.

Often the `IEEEeqnarray` environment (see section 3.5) will yield better results. Consider the following situation:

```
\begin{equation}
  a = b + c + d + e + f
  + g + h + i + j
  + k + l + m + n + o + p
  \label{eq:equation_too_long}
\end{equation}
```

$$
a = b+c+d+e+f+g+h+i+j+k+l+m+n+o+p \tag{3.5}
$$

[6] The `multline`-environment is from `amsmath`.

Here it is actually the RHS that is too long to fit on one line. The `multline` environment creates the following output:

```
\begin{multline}
  a = b + c + d + e + f
  + g + h + i + j \\
  + k + l + m + n + o + p
\end{multline}
```

$$\begin{multline} a = b + c + d + e + f + g + h + i + j \\ + k + l + m + n + o + p \tag{3.6} \end{multline}$$

This is better than (3.5), but it has the disadvantage that the equality sign loses its natural greater importance with respect to the plus operator in front of k. The better solution is provided by the `IEEEeqnarray` environment that will be discussed in detail in Section 3.5.

3.5 Multiple Equations

In the most general situation we have a sequence of several equalities that do not fit onto one line. Here we need to work with vertical alignment in order to keep the array of equations in a nice and readable structure.

Before we offer our suggestions on how to do this, we start with a few bad examples that show the biggest drawbacks of some common solutions.

3.5.1 Problems with Traditional Commands

To group multiple equations the `align` environment[7] could be used:

```
\begin{align}
  a & = b + c \\
  & = d + e
\end{align}
```

$$\begin{align} a & = b + c \tag{3.7} \\ & = d + e \tag{3.8} \end{align}$$

this approach fails once a single line is too long:

```
\begin{align}
  a & = b + c \\
  & = d + e + f + g + h + i
  + j + k + l \nonumber \\
  & + m + n + o \\
  & = p + q + r + s
\end{align}
```

$$\begin{align} a & = b + c \tag{3.9} \\ & = d + e + f + g + h + i + j + k + l \nonumber \\ & + m + n + o \tag{3.10} \\ & = p + q + r + s \tag{3.11} \end{align}$$

[7] The `align`-environment can also be used to group several blocks of equations beside each other. Another excellent use case for the `IEEEeqnarray` environment. Try an argument like `{rCl+rCl}`.

Here $+m$ should be below d and not below the equality sign. A TEXpert will point out that `\mathrel{\phantom{=}} \negmedspace {}`, would add the necessary space in front of `+m+n+o`, but since most users lack that kind of imagination, a simpler solution would be nice.

This is the moment where the `eqnarray` environment bursts onto the scene:

```
\begin{eqnarray}
  a & = & b + c \\
  & = & d + e + f + g + h + i
  + j + k + l \nonumber \\
  && +\: m + n + o \\
  & = & p + q + r + s
\end{eqnarray}
```

$$\begin{array}{rcll} a & = & b+c & (3.12) \\ & = & d+e+f+g+h+i+j+k+l & \\ & & +\,m+n+o & (3.13) \\ & = & p+q+r+s & (3.14) \end{array}$$

It is better but still not optimal. The spaces around the equality signs are too big. Particularly, they are **not** the same as in the `multline` and `equation` environments:

```
\begin{eqnarray}
  a & = & a = a
\end{eqnarray}
```

$$a \quad = \quad a = a \qquad (3.15)$$

...and the expression sometimes overlaps with the equation number even though there would be enough room on the left:

```
\begin{eqnarray}
  a & = & b + c
  \\
  & = & d + e + f + g + h^2
  + i^2 + j
  \label{eq:faultyeqnarray}
\end{eqnarray}
```

$$\begin{array}{rcll} a & = & b+c & (3.16) \\ & = & d+e+f+g+h^2+i^2+j & (3.17) \end{array}$$

While the environment offers a command `\lefteqn` that can be used when the LHS is too long:

```
\begin{eqnarray}
  \lefteqn{a + b + c + d
    + e + f + g + h}\nonumber\\
  & = & i + j + k + l + m
  \\
  & = & n + o + p + q + r + s
\end{eqnarray}
```

$$\begin{array}{rcll} \multicolumn{3}{l}{a+b+c+d+e+f+g+h} & \\ & = & i+j+k+l+m & (3.18) \\ & = & n+o+p+q+r+s & (3.19) \end{array}$$

This is not optimal either as the RHS is too short and the array is not properly centered:

```
\begin{eqnarray}
  \lefteqn{a + b + c + d
    + e + f + g + h}
  \nonumber \\
  & = & i + j
\end{eqnarray}
```

$$
\begin{aligned}
&a+b+c+d+e+f+g+h \\
&\quad = \quad i+j \qquad (3.20)
\end{aligned}
$$

Having badmouthed the competition sufficiently, I can now steer you gently towards the glorious ...

3.5.2 IEEEeqnarray Environment

The `IEEEeqnarray` environment is a very powerful command with many options. Here, we will only introduce its basic functionalities. For more information please refer to the manual.[8]

First of all, in order to be able to use the `IEEEeqnarray` environment one needs to load the package[9] IEEEtrantools. Include the following line in the header of your document:

```
\usepackage{IEEEtrantools}
```

The strength of `IEEEeqnarray` is the ability to specify the number of *columns* in the equation array. Usually, this specification will be `{rCl}`, *i.e.*, three columns, the first column right-justified, the middle one centered with a little more space around it (therefore we specify capital `C` instead of lower-case `c`) and the third column left-justified:

```
\begin{IEEEeqnarray}{rCl}
  a & = & b + c
  \\
  & = & d + e + f + g + h
  + i + j + k \nonumber\\
  && \negmedspace {} + l + m + n + o
  \\
  & = & p + q + r + s
\end{IEEEeqnarray}
```

$$
\begin{aligned}
a &= b+c && (3.21)\\
&= d+e+f+g+h+i+j+k \\
&\quad +l+m+n+o && (3.22)\\
&= p+q+r+s && (3.23)
\end{aligned}
$$

Any number of columns can be specified: `{c}` will give only one column with all entries centered, or `{rCll}` would add a fourth, left-justified column to use for comments. Moreover, beside `l`, `c`, `r`, `L`, `C`, `R` for math mode entries there are also `s`, `t`, `u` for left, centered, and right text mode entries. Additional space can be added with `.` and `/` and `?` in increasing order.[10] Note the spaces around the equality signs in contrast to the space produced by the `eqnarray` environment.

[8]The official manual is called `CTAN://macros/latex/contrib/IEEEtran/IEEEtran_HOWTO.pdf`. The part about `IEEEeqnarray` can be found in Appendix F.

[9]The IEEEtrantools package may not be included in your setup, it can be found on CTAN.

[10]For more spacing types refer to Section 3.9.1.

3.5.3 Common Usage

In the following we will describe how we use `IEEEeqnarray` to solve the most common problems.

If a line overlaps with the equation number as in (3.17), the command

```
\IEEEeqnarraynumspace
```

can be used: it has to be added in the corresponding line and makes sure that the whole equation array is shifted by the size of the equation numbers (the shift depends on the size of the number!): instead of

```
\begin{IEEEeqnarray}{rCl}
  a & = & b + c
  \\
  & = & d + e + f + g + h
  + i + j + k
  \\
  & = & l + m + n
\end{IEEEeqnarray}
```

$$a = b + c \tag{3.24}$$
$$= d + e + f + g + h + i + j + k \tag{3.25}$$
$$= l + m + n \tag{3.26}$$

we get

```
\begin{IEEEeqnarray}{rCl}
  a & = & b + c
  \\
  & = & d + e + f + g + h
  + i + j + k
  \IEEEeqnarraynumspace\\
  & = & l + m + n.
\end{IEEEeqnarray}
```

$$a = b + c \tag{3.27}$$
$$= d + e + f + g + h + i + j + k \tag{3.28}$$
$$= l + m + n. \tag{3.29}$$

If the LHS is too long, as a replacement for the faulty `\lefteqn` command, `IEEEeqnarray` offers the `\IEEEeqnarraymulticol` command which works in all situations:

```
\begin{IEEEeqnarray}{rCl}
  \IEEEeqnarraymulticol{3}{l}{
    a + b + c + d + e + f
    + g + h
  }\nonumber\\ \quad
  & = & i + j
  \\
  & = & k + l + m
\end{IEEEeqnarray}
```

$$a + b + c + d + e + f + g + h$$
$$\quad = i + j \tag{3.30}$$
$$\quad = k + l + m \tag{3.31}$$

The usage is identical to the `\multicolumns` command in the `tabular`-environment. The first argument `{3}` specifies that three columns shall be combined into one which will be left-justified `{l}`.

Note that by inserting \quad commands one can easily adapt the depth of the equation signs,[11] *e.g.*,

```
\begin{IEEEeqnarray}{rCl}
  \IEEEeqnarraymulticol{3}{l}{
    a + b + c + d + e + f
    + g + h
  }\nonumber\\ \qquad\qquad
  & = & i + j
  \\
  & = & k + l + m
\end{IEEEeqnarray}
```

$$
\begin{aligned}
&a+b+c+d+e+f+g+h \\
&\qquad\qquad = i+j && (3.32)\\
&\qquad\qquad = k+l+m && (3.33)
\end{aligned}
$$

If an equation is split into two or more lines, LaTeX interprets the first + or − as a sign instead of operator. Therefore, it is necessary to add an empty group {} before the operator: instead of

```
\begin{IEEEeqnarray}{rCl}
  a & = & b + c
  \\
  & = & d + e + f + g + h
  + i + j + k \nonumber\\
  && + l + m + n + o
  \\
  & = & p + q + r + s
\end{IEEEeqnarray}
```

$$
\begin{aligned}
a &= b+c && (3.34)\\
&= d+e+f+g+h+i+j+k \\
&\quad +l+m+n+o && (3.35)\\
&= p+q+r+s && (3.36)
\end{aligned}
$$

we should write

```
\begin{IEEEeqnarray}{rCl}
  a & = & b + c
  \\
  & = & d + e + f + g + h
  + i + j + k \nonumber\\
  && \negmedspace {} + l + m + n + o
  \\
  & = & p + q + r + s
\end{IEEEeqnarray}
```

$$
\begin{aligned}
a &= b+c && (3.37)\\
&= d+e+f+g+h+i+j+k \\
&\quad {}+l+m+n+o && (3.38)\\
&= p+q+r+s && (3.39)
\end{aligned}
$$

Note the space difference between + and l! The construction `{} + l` forces the +-sign to be an operator rather than just a sign, and the unwanted ensuing space between {} and + is compensated by a negative medium space \negmedspace.

If a particular line should not have an equation number, the number can be suppressed using \nonumber (or \IEEEnonumber). If on such a line a label \label{eq:...} is defined, then this label is passed on to the next

[11] I think that one quad is the distance that looks good for most cases.

equation number that is not suppressed. Place the labels right before the line-break \\ or the next to the equation it belongs to. Apart from improving the readability of the source code this prevents a compilation error when a `\IEEEmulticol` command follows the label-definition.

There also exists a *-version where all equation numbers are suppressed. In this case an equation number can be made to appear using the command `\IEEEyesnumber`:

```
\begin{IEEEeqnarray*}{rCl}
  a & = & b + c \\
  & = & d + e \IEEEyesnumber\\
  & = & f + g
\end{IEEEeqnarray*}
```

$$\begin{aligned} a &= b + c \\ &= d + e \qquad (3.40) \\ &= f + g \end{aligned}$$

Sub-numbers are also easily possible using `\IEEEyessubnumber`:

```
\begin{IEEEeqnarray}{rCl}
  a & = & b + c
  \IEEEyessubnumber\\
  & = & d + e
  \nonumber\\
  & = & f + g
  \IEEEyessubnumber
\end{IEEEeqnarray}
```

$$\begin{aligned} a &= b + c \qquad (3.40a) \\ &= d + e \\ &= f + g \qquad (3.40b) \end{aligned}$$

3.6 Arrays and Matrices

To typeset **arrays**, use the `array` environment. It works in a similar way to the `tabular` environment. The \\ command is used to break the lines:

```
\begin{equation*}
  \mathbf{X} = \left(
    \begin{array}{ccc}
      x_1 & x_2 & \ldots \\
      x_3 & x_4 & \ldots \\
      \vdots & \vdots & \ddots
    \end{array} \right)
\end{equation*}
```

$$\mathbf{X} = \left(\begin{array}{ccc} x_1 & x_2 & \ldots \\ x_3 & x_4 & \ldots \\ \vdots & \vdots & \ddots \end{array} \right)$$

The `array` environment can also be used to typeset piecewise functions by using a “.” as an invisible `\right` delimiter:

```
\begin{equation*}
  |x| = \left\{
    \begin{array}{rl}
      -x & \text{if } x < 0,\\
      0 & \text{if } x = 0,\\
      x & \text{if } x > 0.
    \end{array} \right.
\end{equation*}
```

$$|x| = \left\{ \begin{array}{rl} -x & \text{if } x < 0,\\ 0 & \text{if } x = 0,\\ x & \text{if } x > 0. \end{array} \right.$$

The `cases` environment from amsmath simplifies the syntax, so it is worth a look:

```
  \begin{equation*}
    |x| =
    \begin{cases}
      -x & \text{if } x < 0,\\
      0 & \text{if } x = 0,\\
      x & \text{if } x > 0.
    \end{cases}
\end{equation*}
```

$$|x| = \begin{cases} -x & \text{if } x < 0,\\ 0 & \text{if } x = 0,\\ x & \text{if } x > 0. \end{cases}$$

Matrices can be typeset by `array`, but amsmath provides a better solution using the different `matrix` environments. There are six versions with different delimiters: `matrix` (none), `pmatrix` (, `bmatrix` [, `Bmatrix` {, `vmatrix` | and `Vmatrix` ‖. You don't have to specify the number of columns as with `array`. The maximum number is 10, but it is customisable (though it is not very often you need 10 columns!):

```
\begin{equation*}
  \begin{matrix}
    1 & 2 \\
    3 & 4
  \end{matrix} \qquad
  \begin{bmatrix}
    p_{11} & p_{12} & \ldots
    & p_{1n} \\
    p_{21} & p_{22} & \ldots
    & p_{2n} \\
    \vdots & \vdots & \ddots
    & \vdots \\
    p_{m1} & p_{m2} & \ldots
    & p_{mn}
  \end{bmatrix}
\end{equation*}
```

$$\begin{matrix} 1 & 2 \\ 3 & 4 \end{matrix} \qquad \begin{bmatrix} p_{11} & p_{12} & \ldots & p_{1n} \\ p_{21} & p_{22} & \ldots & p_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{m1} & p_{m2} & \ldots & p_{mn} \end{bmatrix}$$

3.7 Spacing in Math Mode

If the spacing within formulae chosen by LaTeX is not satisfactory, it can be adjusted by inserting special spacing commands: `\,` for $\frac{3}{18}$ quad (␣), `\:` for $\frac{4}{18}$ quad (␣) and `\;` for $\frac{5}{18}$ quad (␣). The escaped space character `\␣` generates a medium sized space comparable to the interword spacing and `\quad` (␣) and `\qquad` (␣) produce large spaces. The size of a `\quad` corresponds to the width of the character 'M' of the current font. `\!` produces a negative space of $-\frac{3}{18}$ quad (−␣).

```
\begin{equation*}
  \int_1^2 \ln x \mathrm{d}x
  \qquad
  \int_1^2 \ln x \,\mathrm{d}x
\end{equation*}
```

$$\int_1^2 \ln x \mathrm{d}x \qquad \int_1^2 \ln x \,\mathrm{d}x$$

Note that 'd' in the differential is conventionally set in roman. In the next example, we define a new command `\ud` (upright d) which produces "␣d" (notice the spacing ␣ before the d), so we don't have to write it every time. The `\newcommand` is placed in the preamble.

```
\newcommand{\ud}{\,\mathrm{d}}

\begin{equation*}
 \int_a^b f(x)\ud x
\end{equation*}
```

$$\int_a^b f(x)\,\mathrm{d}x$$

If you want to typeset multiple integrals, you'll discover that the spacing between the integrals is too wide. You can correct it using `\!`, but amsmath provides an easier way for fine-tuning the spacing, namely the `\iint`, `\iiint`, `\iiiint`, and `\idotsint` commands.

```
\newcommand{\ud}{\,\mathrm{d}}

\begin{IEEEeqnarray*}{c}
  \int\int f(x)g(y)
                  \ud x \ud y \\
  \int\!\!\!\int
        f(x)g(y) \ud x \ud y \\
  \iint f(x)g(y)  \ud x \ud y
\end{IEEEeqnarray*}
```

$$\int\int f(x)g(y)\,\mathrm{d}x\,\mathrm{d}y$$
$$\int\!\!\!\int f(x)g(y)\,\mathrm{d}x\,\mathrm{d}y$$
$$\iint f(x)g(y)\,\mathrm{d}x\,\mathrm{d}y$$

See the electronic document `testmath.tex` (distributed with AMS-LaTeX) or Chapter 8 of *The LaTeX Companion* [3] for further details.

3.7.1 Phantoms

When vertically aligning text using ^ and _ LaTeX is sometimes just a little too helpful. Using the \phantom command you can reserve space for characters that do not show up in the final output. The easiest way to understand this is to look at an example:

```
\begin{equation*}
{}^{14}_{6}\text{C}
\qquad \text{versus} \qquad
{}^{14}_{\phantom{1}6}\text{C}
\end{equation*}
```

$$
{}^{14}_{6}\text{C}
\qquad \text{versus} \qquad
{}^{14}_{6}\text{C}
$$

If you want to typeset a lot of isotopes as in the example, the mhchem package is very useful for typesetting isotopes and chemical formulae too.

3.8 Fiddling with the Math Fonts

Different math fonts are listed on Table 3.14 on page 69.

```
$\Re \qquad
 \mathcal{R} \qquad
 \mathfrak{R} \qquad
 \mathbb{R} \qquad $
```

$\Re \qquad \mathcal{R} \qquad \mathfrak{R} \qquad \mathbb{R} \qquad$

The last two require amssymb or amsfonts.

Sometimes you need to tell LaTeX the correct font size. In math mode, this is set with the following four commands:

\displaystyle (123), \textstyle (123), \scriptstyle ($\scriptstyle 123$) and \scriptscriptstyle ($\scriptscriptstyle 123$).

If $\sum$ is placed in a fraction, it'll be typeset in text style unless you tell LaTeX otherwise:

```
\begin{equation*}
 P = \frac{\displaystyle{
   \sum_{i=1}^n (x_i- x)
   (y_i- y)}}
   {\displaystyle{\left[
   \sum_{i=1}^n(x_i-x)^2
   \sum_{i=1}^n(y_i- y)^2
   \right]^{1/2}}}
\end{equation*}
```

$$
P = \frac{\displaystyle{\sum_{i=1}^n (x_i- x)(y_i- y)}}{\displaystyle{\left[\sum_{i=1}^n(x_i-x)^2\sum_{i=1}^n(y_i- y)^2\right]^{1/2}}}
$$

Changing styles generally affects the way big operators and limits are displayed.

3.8.1 Bold Symbols

It is quite difficult to get bold symbols in LaTeX; this is probably intentional as amateur typesetters tend to overuse them. The font change command `\mathbf` gives bold letters, but these are roman (upright) whereas mathematical symbols are normally italic, and furthermore it doesn't work on lower case Greek letters. There is a `\boldmath` command, but *this can only be used outside math mode.* It works for symbols too, though:

```
$\mu, M \qquad
\mathbf{\mu}, \mathbf{M}$
\qquad \boldmath{$\mu, M$}
```

$\mu, M \qquad \mu, \mathbf{M} \qquad \boldsymbol{\mu}, \boldsymbol{M}$

The package amsbsy (included by amsmath) as well as the package bm from the `tools` bundle make this much easier as they include a `\boldsymbol` command:

```
$\mu, M \qquad
\boldsymbol{\mu}, \boldsymbol{M}$
```

$\mu, M \qquad \boldsymbol{\mu}, \boldsymbol{M}$

3.9 Theorems, Lemmas, …

When writing mathematical documents, you probably need a way to typeset "Lemmas", "Definitions", "Axioms" and similar structures.

`\newtheorem{`*name*`}[`*counter*`]{`*text*`}[`*section*`]`

The *name* argument is a short keyword used to identify the "theorem". With the *text* argument you define the actual name of the "theorem", which will be printed in the final document.

The arguments in square brackets are optional. They are both used to specify the numbering used on the "theorem". Use the *counter* argument to specify the *name* of a previously declared "theorem". The new "theorem" will then be numbered in the same sequence. The *section* argument allows you to specify the sectional unit within which the "theorem" should get its numbers.

After executing the `\newtheorem` command in the preamble of your document, you can use the following command within the document.

`\begin{`*name*`}[`*text*`]`
This is my interesting theorem
`\end{`*name*`}`

The amsthm package (part of AMS-LaTeX) provides the `\theoremstyle{`*style*`}` command which lets you define what the theorem is all about by picking from three predefined styles: `definition` (fat title, roman body), `plain` (fat title, italic body) or `remark` (italic title, roman body).

This should be enough theory. The following examples should remove any remaining doubt, and make it clear that the `\newtheorem` environment is way too complex to understand.

First define the theorems:

```
\theoremstyle{definition} \newtheorem{law}{Law}
\theoremstyle{plain}      \newtheorem{jury}[law]{Jury}
\theoremstyle{remark}     \newtheorem*{marg}{Margaret}
```

```
\begin{law} \label{law:box}
Don't hide in the witness box
\end{law}
\begin{jury}[The Twelve]
It could be you! So beware and
see law~\ref{law:box}.\end{jury}
\begin{jury}
You will disregard the last
statement.\end{jury}
\begin{marg}No, No, No\end{marg}
\begin{marg}Denis!\end{marg}
```

Law 1. Don't hide in the witness box

Jury 2 (The Twelve)**.** *It could be you! So beware and see law 1.*

Jury 3. *You will disregard the last statement.*

Margaret. No, No, No

Margaret. Denis!

The "Jury" theorem uses the same counter as the "Law" theorem, so it gets a number that is in sequence with the other "Laws". The argument in square brackets is used to specify a title or something similar for the theorem.

```
\newtheorem{mur}{Murphy}[section]

\begin{mur} If there are two or
more ways to do something, and
one of those ways can result in
a catastrophe, then someone
will do it.\end{mur}
```

Murphy 3.9.1. If there are two or more ways to do something, and one of those ways can result in a catastrophe, then someone will do it.

The "Murphy" theorem gets a number that is linked to the number of the current section. You could also use another unit, for example chapter or subsection.

If you want to customize your theorems down to the last dot, the ntheorem package offers a plethora of options.

3.9.1 Proofs and End-of-Proof Symbol

The amsthm package also provides the `proof` environment.

```
\begin{proof}
 Trivial, use
 \begin{equation*}
   E=mc^2.
 \end{equation*}
\end{proof}
```

Proof. Trivial, use

$$E = mc^2.$$

□

With the command `\qedhere` you can move the 'end of proof' symbol around for situations where it would end up alone on a line.

```
\begin{proof}
 Trivial, use
 \begin{equation*}
   E=mc^2. \qedhere
 \end{equation*}
\end{proof}
```

Proof. Trivial, use

$$E = mc^2. \qquad \square$$

Unfortunately, this correction does not work for `IEEEeqnarray`:

```
\begin{proof}
  This is a proof that ends
  with an equation array:
  \begin{IEEEeqnarray*}{rCl}
    a & = & b + c \\
    & = & d + e. \qedhere
  \end{IEEEeqnarray*}
\end{proof}
```

Proof. This is a proof that ends with an equation array:

$$\begin{aligned} a &= b + c \\ &= d + e. \quad \square \end{aligned}$$

The reason for this is the internal structure of `IEEEeqnarray`: it always puts two invisible columns at both sides of the array that only contain a stretchable space. By this `IEEEeqnarray` ensures that the equation array is horizontally centered. The `\qedhere` command should actually be put *outside* this stretchable space, but this does not happen as these columns are invisible to the user.

There is a very simple remedy. Define the stretching explicitly!

```
\begin{proof}
  This is a proof that ends
  with an equation array:
  \begin{IEEEeqnarray*}{+rCl+x*}
    a & = & b + c \\
    & = & d + e. & \qedhere
  \end{IEEEeqnarray*}
\end{proof}
```

Proof. This is a proof that ends with an equation array:

$$\begin{aligned} a &= b + c \\ &= d + e. \qquad\qquad \square \end{aligned}$$

Note that the `+` in `{+rCl+x*}` denotes stretchable spaces, one on the left of the equations (which, if not specified, will be done automatically by `IEEEeqnarray`!) and one on the right of the equations. But now on the right, *after* the stretching column, we add an empty column `x`. This column will only be needed on the last line if the `\qedhere` command is put there. Finally, we specify a `*`. This is a null-space that prevents `IEEEeqnarray` from adding another unwanted +-space!

In the case of equation numbering, there is a similar problem. Comparing

```
\begin{proof}
  This is a proof that ends
  with a numbered equation:
  \begin{equation}
    a = b + c.
  \end{equation}
\end{proof}
```

Proof. This is a proof that ends with a numbered equation:

$$a = b + c. \tag{3.41}$$

□

with

```
\begin{proof}
  This is a proof that ends
  with a numbered equation:
  \begin{equation}
    a = b + c. \qedhere
  \end{equation}
\end{proof}
```

Proof. This is a proof that ends with a numbered equation:

$$a = b + c. \tag{3.42}$$
□

you notice that in the (correct) second version the □ is much closer to the equation than in the first version.

Similarly, the correct way of putting the QED-symbol at the end of an equation array is as follows:

```
\begin{proof}
  This is a proof that ends
  with an equation array:
  \begin{IEEEeqnarray}{+rCl+x*}
    a & = & b + c \\
    & = & d + e. \\
    &&& \qedhere\nonumber
  \end{IEEEeqnarray}
\end{proof}
```

Proof. This is a proof that ends with an equation array:

$$a = b + c \tag{3.43}$$
$$= d + e. \tag{3.44}$$

□

which contrasts with

```
\begin{proof}
  This is a proof that ends
  with an equation array:
  \begin{IEEEeqnarray}{rCl}
    a & = & b + c \\
    & = & d + e.
  \end{IEEEeqnarray}
\end{proof}
```

Proof. This is a proof that ends with an equation array:

$$a = b + c \tag{3.45}$$
$$= d + e. \tag{3.46}$$

□

3.10 List of Mathematical Symbols

The following tables demonstrate all the symbols normally accessible from *math mode*.

Note that some tables show symbols only accessible after loading the amssymb package in the preamble of your document[12]. If the $\mathcal{AMS}$ package and fonts are not installed on your system, have a look at `CTAN:pkg/amslatex`. An even more comprehensive list of symbols can be found at `CTAN:info/symbols/comprehensive`.

Table 3.1: Math Mode Accents.

$\hat{a}$	`\hat{a}`	$\check{a}$	`\check{a}`	$\tilde{a}$	`\tilde{a}`
$\grave{a}$	`\grave{a}`	$\dot{a}$	`\dot{a}`	$\ddot{a}$	`\ddot{a}`
$\bar{a}$	`\bar{a}`	$\vec{a}$	`\vec{a}`	$\widehat{AAA}$	`\widehat{AAA}`
$\acute{a}$	`\acute{a}`	$\breve{a}$	`\breve{a}`	$\widetilde{AAA}$	`\widetilde{AAA}`
$\mathring{a}$	`\mathring{a}`				

Table 3.2: Greek Letters.

There is no uppercase of some of the letters like `\Alpha`, `\Beta` and so on, because they look the same as normal roman letters: A, B...

α	`\alpha`	θ	`\theta`	o	`o`	υ	`\upsilon`
β	`\beta`	ϑ	`\vartheta`	π	`\pi`	ϕ	`\phi`
γ	`\gamma`	ι	`\iota`	ϖ	`\varpi`	φ	`\varphi`
δ	`\delta`	κ	`\kappa`	ρ	`\rho`	χ	`\chi`
ϵ	`\epsilon`	λ	`\lambda`	ϱ	`\varrho`	ψ	`\psi`
ε	`\varepsilon`	μ	`\mu`	σ	`\sigma`	ω	`\omega`
ζ	`\zeta`	ν	`\nu`	ς	`\varsigma`		
η	`\eta`	ξ	`\xi`	τ	`\tau`		
Γ	`\Gamma`	Λ	`\Lambda`	Σ	`\Sigma`	Ψ	`\Psi`
Δ	`\Delta`	Ξ	`\Xi`	Υ	`\Upsilon`	Ω	`\Omega`
Θ	`\Theta`	Π	`\Pi`	Φ	`\Phi`		

[12]The tables were derived from `symbols.tex` by David Carlisle and subsequently changed extensively as suggested by Josef Tkadlec.

Table 3.3: Binary Relations.

You can negate the following symbols by prefixing them with a `\not` command.

$<$	`<`	$>$	`>`	$=$	`=`
$\leq$	`\leq` or `\le`	$\geq$	`\geq` or `\ge`	$\equiv$	`\equiv`
$\ll$	`\ll`	$\gg$	`\gg`	$\doteq$	`\doteq`
$\prec$	`\prec`	$\succ$	`\succ`	$\sim$	`\sim`
$\preceq$	`\preceq`	$\succeq$	`\succeq`	$\simeq$	`\simeq`
$\subset$	`\subset`	$\supset$	`\supset`	$\approx$	`\approx`
$\subseteq$	`\subseteq`	$\supseteq$	`\supseteq`	$\cong$	`\cong`
$\sqsubset$	`\sqsubset` [a]	$\sqsupset$	`\sqsupset` [a]	$\Join$	`\Join` [a]
$\sqsubseteq$	`\sqsubseteq`	$\sqsupseteq$	`\sqsupseteq`	$\bowtie$	`\bowtie`
$\in$	`\in`	$\ni$	`\ni` , `\owns`	$\propto$	`\propto`
$\vdash$	`\vdash`	$\dashv$	`\dashv`	$\models$	`\models`
$\mid$	`\mid`	$\parallel$	`\parallel`	$\perp$	`\perp`
$\smile$	`\smile`	$\frown$	`\frown`	$\asymp$	`\asymp`
$:$	`:`	$\notin$	`\notin`	$\neq$	`\neq` or `\ne`

[a]Use the latexsym package to access this symbol

Table 3.4: Binary Operators.

$+$	`+`	$-$	`-`		
$\pm$	`\pm`	$\mp$	`\mp`	$\triangleleft$	`\triangleleft`
$\cdot$	`\cdot`	$\div$	`\div`	$\triangleright$	`\triangleright`
$\times$	`\times`	$\setminus$	`\setminus`	$\star$	`\star`
$\cup$	`\cup`	$\cap$	`\cap`	$\ast$	`\ast`
$\sqcup$	`\sqcup`	$\sqcap$	`\sqcap`	$\circ$	`\circ`
$\vee$	`\vee` , `\lor`	$\wedge$	`\wedge` , `\land`	$\bullet$	`\bullet`
$\oplus$	`\oplus`	$\ominus$	`\ominus`	$\diamond$	`\diamond`
$\odot$	`\odot`	$\oslash$	`\oslash`	$\uplus$	`\uplus`
$\otimes$	`\otimes`	$\bigcirc$	`\bigcirc`	$\amalg$	`\amalg`
$\bigtriangleup$	`\bigtriangleup`	$\bigtriangledown$	`\bigtriangledown`	$\dagger$	`\dagger`
$\lhd$	`\lhd` [a]	$\rhd$	`\rhd` [a]	$\ddagger$	`\ddagger`
$\unlhd$	`\unlhd` [a]	$\unrhd$	`\unrhd` [a]	$\wr$	`\wr`

Table 3.5: BIG Operators.

$\sum$	`\sum`	$\bigcup$	`\bigcup`	$\bigvee$	`\bigvee`
$\prod$	`\prod`	$\bigcap$	`\bigcap`	$\bigwedge$	`\bigwedge`
$\coprod$	`\coprod`	$\bigsqcup$	`\bigsqcup`	$\biguplus$	`\biguplus`
$\int$	`\int`	$\oint$	`\oint`	$\bigodot$	`\bigodot`
$\bigoplus$	`\bigoplus`	$\bigotimes$	`\bigotimes`		

Table 3.6: Arrows.

$\leftarrow$	`\leftarrow` or `\gets`	$\longleftarrow$	`\longleftarrow`
$\rightarrow$	`\rightarrow` or `\to`	$\longrightarrow$	`\longrightarrow`
$\leftrightarrow$	`\leftrightarrow`	$\longleftrightarrow$	`\longleftrightarrow`
$\Leftarrow$	`\Leftarrow`	$\Longleftarrow$	`\Longleftarrow`
$\Rightarrow$	`\Rightarrow`	$\Longrightarrow$	`\Longrightarrow`
$\Leftrightarrow$	`\Leftrightarrow`	$\Longleftrightarrow$	`\Longleftrightarrow`
$\mapsto$	`\mapsto`	$\longmapsto$	`\longmapsto`
$\hookleftarrow$	`\hookleftarrow`	$\hookrightarrow$	`\hookrightarrow`
$\leftharpoonup$	`\leftharpoonup`	$\rightharpoonup$	`\rightharpoonup`
$\leftharpoondown$	`\leftharpoondown`	$\rightharpoondown$	`\rightharpoondown`
$\rightleftharpoons$	`\rightleftharpoons`	$\iff$	`\iff` (bigger spaces)
$\uparrow$	`\uparrow`	$\downarrow$	`\downarrow`
$\updownarrow$	`\updownarrow`	$\Uparrow$	`\Uparrow`
$\Downarrow$	`\Downarrow`	$\Updownarrow$	`\Updownarrow`
$\nearrow$	`\nearrow`	$\searrow$	`\searrow`
$\swarrow$	`\swarrow`	$\nwarrow$	`\nwarrow`
$\leadsto$	`\leadsto` [a]		

[a] Use the latexsym package to access this symbol

Table 3.7: Arrows as Accents.

$\overrightarrow{AB}$	`\overrightarrow{AB}`	$\underrightarrow{AB}$	`\underrightarrow{AB}`
$\overleftarrow{AB}$	`\overleftarrow{AB}`	$\underleftarrow{AB}$	`\underleftarrow{AB}`
$\overleftrightarrow{AB}$	`\overleftrightarrow{AB}`	$\underleftrightarrow{AB}$	`\underleftrightarrow{AB}`

Table 3.8: Delimiters.

<table>
<tr><td>$($</td><td>(</td><td>$)$</td><td>)</td><td>$\uparrow$</td><td>\uparrow</td></tr>
<tr><td>$[$</td><td>[or \lbrack</td><td>$]$</td><td>] or \rbrack</td><td>$\downarrow$</td><td>\downarrow</td></tr>
<tr><td>$\{$</td><td>\{ or \lbrace</td><td>$\}$</td><td>\} or \rbrace</td><td>$\updownarrow$</td><td>\updownarrow</td></tr>
<tr><td>$\langle$</td><td>\langle</td><td>$\rangle$</td><td>\rangle</td><td>$\Uparrow$</td><td>\Uparrow</td></tr>
<tr><td>$\vert$</td><td>| or \vert</td><td>$\Vert$</td><td>\| or \Vert</td><td>$\Downarrow$</td><td>\Downarrow</td></tr>
<tr><td>$/$</td><td>/</td><td>$\backslash$</td><td>\backslash</td><td>$\Updownarrow$</td><td>\Updownarrow</td></tr>
<tr><td>$\lfloor$</td><td>\lfloor</td><td>$\rfloor$</td><td>\rfloor</td><td></td><td></td></tr>
<tr><td>$\rceil$</td><td>\rceil</td><td>$\lceil$</td><td>\lceil</td><td></td><td></td></tr>
</table>

Table 3.9: Large Delimiters.

$\lgroup$	\lgroup	$\rgroup$	\rgroup	$\lmoustache$	\lmoustache
$\arrowvert$	\arrowvert	$\Arrowvert$	\Arrowvert	$\bracevert$	\bracevert
$\rmoustache$	\rmoustache				

Table 3.10: Miscellaneous Symbols.

$\dots$	\dots	$\cdots$	\cdots	$\vdots$	\vdots	$\ddots$	\ddots
$\hbar$	\hbar	$\imath$	\imath	$\jmath$	\jmath	ℓ	\ell
$\Re$	\Re	$\Im$	\Im	$\aleph$	\aleph	$\wp$	\wp
$\forall$	\forall	$\exists$	\exists	$\mho$	\mho [a]	∂	\partial
$'$	'	$\prime$	\prime	$\emptyset$	\emptyset	∞	\infty
∇	\nabla	$\triangle$	\triangle	$\Box$	\Box [a]	$\Diamond$	\Diamond [a]
$\bot$	\bot	$\top$	\top	$\angle$	\angle	$\surd$	\surd
$\diamondsuit$	\diamondsuit	$\heartsuit$	\heartsuit	$\clubsuit$	\clubsuit	$\spadesuit$	\spadesuit
$\neg$	\neg or \lnot	$\flat$	\flat	$\natural$	\natural	$\sharp$	\sharp

[a]Use the latexsym package to access this symbol

Table 3.11: Non-Mathematical Symbols.

These symbols can also be used in text mode.

†	\dag	§	\S	©	\copyright	®	\textregistered
‡	\ddag	¶	\P	£	\pounds	%	\%

Table 3.12: $\mathcal{AMS}$ Delimiters.

$\ulcorner$	\ulcorner	$\urcorner$	\urcorner	$\llcorner$	\llcorner	$\lrcorner$	\lrcorner
$\lvert$	\lvert	$\rvert$	\rvert	$\lVert$	\lVert	$\rVert$	\rVert

Table 3.13: $\mathcal{AMS}$ Greek and Hebrew.

$\digamma$	\digamma	$\varkappa$	\varkappa	$\beth$	\beth	$\gimel$	\gimel	$\daleth$	\daleth

Table 3.14: Math Alphabets.

See Table 6.4 on 109 for other math fonts.

Example	Command	Required package
$\mathrm{ABCDE abcde 1234}$	\mathrm{ABCDE abcde 1234}	
$\mathit{ABCDE abcde 1234}$	\mathit{ABCDE abcde 1234}	
$\mathnormal{ABCDE abcde 1234}$	\mathnormal{ABCDE abcde 1234}	
$\mathcal{ABCDE}$	\mathcal{ABCDE abcde 1234}	
$\mathscr{ABCDE}$	\mathscr{ABCDE abcde 1234}	mathrsfs
$\mathfrak{ABCDE abcde 1234}$	\mathfrak{ABCDE abcde 1234}	amsfonts or amssymb
$\mathbb{ABCDE abcde 1234}$	\mathbb{ABCDE abcde 1234}	amsfonts or amssymb

Table 3.15: $\mathcal{AMS}$ Binary Operators.

$\dotplus$	\dotplus	$\centerdot$	\centerdot		
$\ltimes$	\ltimes	$\rtimes$	\rtimes	$\divideontimes$	\divideontimes
$\doublecup$	\doublecup	$\doublecap$	\doublecap	$\smallsetminus$	\smallsetminus
$\veebar$	\veebar	$\barwedge$	\barwedge	$\doublebarwedge$	\doublebarwedge
$\boxplus$	\boxplus	$\boxminus$	\boxminus	$\circleddash$	\circleddash
$\boxtimes$	\boxtimes	$\boxdot$	\boxdot	$\circledcirc$	\circledcirc
$\intercal$	\intercal	$\circledast$	\circledast	$\rightthreetimes$	\rightthreetimes
$\curlyvee$	\curlyvee	$\curlywedge$	\curlywedge	$\leftthreetimes$	\leftthreetimes

Table 3.16: $\mathcal{A}_{\mathcal{M}}\mathcal{S}$ Binary Relations.

$\lessdot$	\lessdot	$\gtrdot$	\gtrdot	$\doteqdot$	\doteqdot
$\leqslant$	\leqslant	$\geqslant$	\geqslant	$\risingdotseq$	\risingdotseq
$\eqslantless$	\eqslantless	$\eqslantgtr$	\eqslantgtr	$\fallingdotseq$	\fallingdotseq
$\leqq$	\leqq	$\geqq$	\geqq	$\eqcirc$	\eqcirc
$\lll$	\lll or \llless	$\ggg$	\ggg	$\circeq$	\circeq
$\lesssim$	\lesssim	$\gtrsim$	\gtrsim	$\triangleq$	\triangleq
$\lessapprox$	\lessapprox	$\gtrapprox$	\gtrapprox	$\bumpeq$	\bumpeq
$\lessgtr$	\lessgtr	$\gtrless$	\gtrless	$\Bumpeq$	\Bumpeq
$\lesseqgtr$	\lesseqgtr	$\gtreqless$	\gtreqless	$\thicksim$	\thicksim
$\lesseqqgtr$	\lesseqqgtr	$\gtreqqless$	\gtreqqless	$\thickapprox$	\thickapprox
$\preccurlyeq$	\preccurlyeq	$\succcurlyeq$	\succcurlyeq	$\approxeq$	\approxeq
$\curlyeqprec$	\curlyeqprec	$\curlyeqsucc$	\curlyeqsucc	$\backsim$	\backsim
$\precsim$	\precsim	$\succsim$	\succsim	$\backsimeq$	\backsimeq
$\precapprox$	\precapprox	$\succapprox$	\succapprox	$\vDash$	\vDash
$\subseteqq$	\subseteqq	$\supseteqq$	\supseteqq	$\Vdash$	\Vdash
$\shortparallel$	\shortparallel	$\Supset$	\Supset	$\Vvdash$	\Vvdash
$\blacktriangleleft$	\blacktriangleleft	$\sqsupset$	\sqsupset	$\backepsilon$	\backepsilon
$\vartriangleright$	\vartriangleright	$\because$	\because	$\varpropto$	\varpropto
$\blacktriangleright$	\blacktriangleright	$\Subset$	\Subset	$\between$	\between
$\trianglerighteq$	\trianglerighteq	$\smallfrown$	\smallfrown	$\pitchfork$	\pitchfork
$\vartriangleleft$	\vartriangleleft	$\shortmid$	\shortmid	$\smallsmile$	\smallsmile
$\trianglelefteq$	\trianglelefteq	$\therefore$	\therefore	$\sqsubset$	\sqsubset

Table 3.17: $\mathcal{AMS}$ Arrows.

$\dashleftarrow$	`\dashleftarrow`	$\dashrightarrow$	`\dashrightarrow`
$\leftleftarrows$	`\leftleftarrows`	$\rightrightarrows$	`\rightrightarrows`
$\leftrightarrows$	`\leftrightarrows`	$\rightleftarrows$	`\rightleftarrows`
$\Lleftarrow$	`\Lleftarrow`	$\Rrightarrow$	`\Rrightarrow`
$\twoheadleftarrow$	`\twoheadleftarrow`	$\twoheadrightarrow$	`\twoheadrightarrow`
$\leftarrowtail$	`\leftarrowtail`	$\rightarrowtail$	`\rightarrowtail`
$\leftrightharpoons$	`\leftrightharpoons`	$\rightleftharpoons$	`\rightleftharpoons`
$\Lsh$	`\Lsh`	$\Rsh$	`\Rsh`
$\looparrowleft$	`\looparrowleft`	$\looparrowright$	`\looparrowright`
$\curvearrowleft$	`\curvearrowleft`	$\curvearrowright$	`\curvearrowright`
$\circlearrowleft$	`\circlearrowleft`	$\circlearrowright$	`\circlearrowright`
$\multimap$	`\multimap`	$\upuparrows$	`\upuparrows`
$\downdownarrows$	`\downdownarrows`	$\upharpoonleft$	`\upharpoonleft`
$\upharpoonright$	`\upharpoonright`	$\downharpoonright$	`\downharpoonright`
$\rightsquigarrow$	`\rightsquigarrow`	$\leftrightsquigarrow$	`\leftrightsquigarrow`

Table 3.18: $\mathcal{AMS}$ Negated Binary Relations and Arrows.

$\nless$	`\nless`	$\ngtr$	`\ngtr`	$\varsubsetneqq$	`\varsubsetneqq`
$\lneq$	`\lneq`	$\gneq$	`\gneq`	$\varsupsetneqq$	`\varsupsetneqq`
$\nleq$	`\nleq`	$\ngeq$	`\ngeq`	$\nsubseteqq$	`\nsubseteqq`
$\nleqslant$	`\nleqslant`	$\ngeqslant$	`\ngeqslant`	$\nsupseteqq$	`\nsupseteqq`
$\lneqq$	`\lneqq`	$\gneqq$	`\gneqq`	$\nmid$	`\nmid`
$\lvertneqq$	`\lvertneqq`	$\gvertneqq$	`\gvertneqq`	$\nparallel$	`\nparallel`
$\nleqq$	`\nleqq`	$\ngeqq$	`\ngeqq`	$\nshortmid$	`\nshortmid`
$\lnsim$	`\lnsim`	$\gnsim$	`\gnsim`	$\nshortparallel$	`\nshortparallel`
$\lnapprox$	`\lnapprox`	$\gnapprox$	`\gnapprox`	$\nsim$	`\nsim`
$\nprec$	`\nprec`	$\nsucc$	`\nsucc`	$\ncong$	`\ncong`
$\npreceq$	`\npreceq`	$\nsucceq$	`\nsucceq`	$\nvdash$	`\nvdash`
$\precneqq$	`\precneqq`	$\succneqq$	`\succneqq`	$\nvDash$	`\nvDash`
$\precnsim$	`\precnsim`	$\succnsim$	`\succnsim`	$\nVdash$	`\nVdash`
$\precnapprox$	`\precnapprox`	$\succnapprox$	`\succnapprox`	$\nVDash$	`\nVDash`
$\subsetneq$	`\subsetneq`	$\supsetneq$	`\supsetneq`	$\ntriangleleft$	`\ntriangleleft`
$\varsubsetneq$	`\varsubsetneq`	$\varsupsetneq$	`\varsupsetneq`	$\ntriangleright$	`\ntriangleright`
$\nsubseteq$	`\nsubseteq`	$\nsupseteq$	`\nsupseteq`	$\ntrianglelefteq$	`\ntrianglelefteq`
$\subsetneqq$	`\subsetneqq`	$\supsetneqq$	`\supsetneqq`	$\ntrianglerighteq$	`\ntrianglerighteq`
$\nleftarrow$	`\nleftarrow`	$\nrightarrow$	`\nrightarrow`	$\nleftrightarrow$	`\nleftrightarrow`
$\nLeftarrow$	`\nLeftarrow`	$\nRightarrow$	`\nRightarrow`	$\nLeftrightarrow$	`\nLeftrightarrow`

Table 3.19: $\mathcal{AMS}$ Miscellaneous.

$\hbar$	`\hbar`	$\hslash$	`\hslash`	$\Bbbk$	`\Bbbk`
$\square$	`\square`	$\blacksquare$	`\blacksquare`	$\circledS$	`\circledS`
$\vartriangle$	`\vartriangle`	$\blacktriangle$	`\blacktriangle`	$\complement$	`\complement`
$\triangledown$	`\triangledown`	$\blacktriangledown$	`\blacktriangledown`	$\Game$	`\Game`
$\lozenge$	`\lozenge`	$\blacklozenge$	`\blacklozenge`	$\bigstar$	`\bigstar`
$\angle$	`\angle`	$\measuredangle$	`\measuredangle`		
$\diagup$	`\diagup`	$\diagdown$	`\diagdown`	$\backprime$	`\backprime`
$\nexists$	`\nexists`	$\Finv$	`\Finv`	$\varnothing$	`\varnothing`
$\eth$	`\eth`	$\sphericalangle$	`\sphericalangle`	$\mho$	`\mho`