Hour 12

Equation Writing II

Writing of basic equations is discussed in Hour 11. Some processes for writing complicated equations are presented here, including the use of mathematical symbols for which special commands are required (such commands are listed in Appendix A on page 247).

12.1 Texts and Blank Space in Math-Mode

Every character in math-mode is treated as a variable and it is printed in math-mode (similar to italic fonts) without any gap between two characters. Normal texts with usual inter-word spacing can be printed in math-mode through the **\mbox{}**, **\text{}** and **\mathrm{}** commands. The commands and fonts discussed in §2.2 on page 11 are also permitted for printing normal texts in math-mode. On the other hand, **~**, **\,**, **\quad**, **\qquad**, **\enspace** and **\hspace{}** can be used in math-mode for generating blank space of different sizes.

Applications of some of the above commands are shown in Table 12.1 on the next page (the same can be found in other sections of the book). In the first example, **\mbox{}** (function of **\text{}** is the same) is used for printing in-line normal texts and **\mathrm{}** for printing a superscript in normal fonts, while **\enspace** is used for maintaining some gap prior to the period mark (the arguments of **\mbox{}** and **\text{}** are in text-mode, while that of **\mathrm{}** is in math-mode)[1]. In the second example in Table 12.1, an array of equations is produced through the **array** environment with alignment at three places. In this example, **\mbox{\boldmath{$**x**$}}** is used for printing x as $\boldsymbol{x}$ (just **\boldmath{**x**}** would print x as x).

[1]The **\mbox{}**, **\text{}** and **\mathrm{}** commands can be used for producing normal texts with usual inter-word spacing in math-modes, where **\mbox{}** and **\text{}** process their arguments in text-mode while **\mathrm{}** processes its argument in math-mode.

D. Datta, *LaTeX in 24 Hours: A Practical Guide for Scientific Writing*,
DOI 10.1007/978-3-319-47831-9_12

Table 12.1 Normal texts and gap in math-mode

LaTeX input	Output
`\begin{equation*} \mbox{Updated value}\quad x = x^\mathrm{low} + yd \enspace . \end{equation*}`	$\text{Updated value}\quad x = x^{\mathrm{low}} + yd \enspace .$
`\begin{equation*} \begin{array}{lll} \mbox{Minimize} & f(\mbox{\boldmath{$x$}}) &\\ \mbox{Subject to} & g_i(\mbox{\boldmath{$x$}}) \leq 0~; & i=1,\ldots,m\\ & h_k(\mbox{\boldmath{$x$}}) = 0~; & k=1,\ldots,p\\ & x_j \geq 0~; & j=1,\ldots,n \end{array} \end{equation*}`	$\begin{array}{lll} \mbox{Minimize} & f(\boldsymbol{x}) & \\ \mbox{Subject to} & g_i(\boldsymbol{x}) \leq 0~; & i=1,\ldots,m\\ & h_k(\boldsymbol{x}) = 0~; & k=1,\ldots,p\\ & x_j \geq 0~; & j=1,\ldots,n \end{array}$

12.2 Conditional Expression

A conditional expression is a case, where a parameter or an expression may take different values in different circumstances. Table 12.2 presents three approaches for

Table 12.2 Conditional mathematical expressions in different forms

LaTeX input	Output
`\begin{equation} \sigma(x) = \begin{cases} e^{\phi xy} \sqrt{x}~, & \text{if}~ x\geq 0\\ 0~, & \text{otherwise.} \end{cases} \end{equation}`	$\sigma(x) = \begin{cases} e^{\phi xy} \sqrt{x}~, & \text{if}~ x\geq 0\\ 0~, & \text{otherwise.} \end{cases} \quad (12.1)$
`\begin{equation} \sigma(x) = \left\{\begin{array}{rl} e^{\phi xy} \sqrt{x}~, & \text{if}~ x\geq 0\\ 0~, & \text{otherwise.} \end{array}\right. \end{equation}`	$\sigma(x) = \left\{\begin{array}{rl} e^{\phi xy} \sqrt{x}~, & \text{if}~ x\geq 0\\ 0~, & \text{otherwise.} \end{array}\right. \quad (12.2)$
`\begin{eqnarray} \sigma(x) &=& e^{\phi xy} \sqrt{x}~,\quad \text{if}~ x\geq 0\\ &=& 0~,\quad \text{otherwise.} \end{eqnarray}`	$\sigma(x) = e^{\phi xy} \sqrt{x}~,\quad \text{if}~ x\geq 0 \quad (12.3)$ $= 0~,\quad \text{otherwise.} \quad (12.4)$

printing such conditional expressions. The first approach uses the **cases** environment inside an **equation** environment, in which conditions are preceded by **&** sign for aligning them. In the **cases** environment, all the conditions are enclosed by a left-hand curly brace, and the entire conditional expression is assigned a single serial number. On the other hand, the **array** environment, applied in the second approach in Table 12.2, is more flexible than the **cases** environment. In the **array** environment, different types of auto-sized delimiters can be obtained on either side of an expression. Moreover, different parts of the expression can be aligned differently. Like the **cases** environment, the **array** environment also can assign a serial number to a set of expressions, if it is nested in a math-mode, such as the **equation** environment. If all the conditions of an expression are to be numbered independently, the **enqarray** environment may be used as shown in Table 12.2 as the third approach for printing the given two conditions by aligning them about the '=' sign (similar effect can be obtained by using other environments given in Table 11.9 on page 107). Note that a gap is to be maintained between **\phi** and `xy` in Table 12.2 (refer §1.5.1 on page 5 for detail).

12.3 Evaluation of Functional Values

Evaluation of a function is an important part of mathematics. Table 12.3 shows an

Table 12.3 Evaluation of a function for a given value

LaTeX input	Output

`\begin{equation*}`
`\left. f\left(\frac{x+1}{2}+2\right) \right|_{x=0}=2.5`
`\end{equation*}`
	$\left. f\left(\frac{x+1}{2}+2\right) \right\|_{x=0}=2.5$

example of evaluating $f(x)$ at $x = 0$, where a vertical line of auto-adjusted height is first produced by using the set of **\left.** and **\right|** commands around the functional expression. Then, the given value ($x = 0$) is printed as the suffix of **|** through **|_{x=0}**.

12.4 Splitting an Equation into Multiple Lines*

It is stated in §11.4 on page 104 that the **equation** or **displaymath** environment, or even the **\[\]**-mode, produces a single equation in a single new line. If an equation is long enough to accommodate in a single line, it can be split into multiple lines through the **multline** environment or its starred form **multline***. In these environments, the first split line is left aligned, the last one is right aligned and all other intermediate lines are centered as shown in Table 12.4 on the next page. The **multline** environment assigns a serial number to the equation, while the **multline*** ignores its numbering.

Table 12.4 Splitting an equation into multiple lines through the **multline** environment

LaTeX input	Output

`\begin{multline}`
`5x_1 + 2x_2 + 3x_3 -\\`
`x_4 - 4x_5 + 5x_6 +\\`
`7x_7 + 3x_8 - 6x_9 -\\`
`2x_{10} - 5x_{11} = 7634`
`\end{multline}`
	$$\begin{multline}5x_1 + 2x_2 + 3x_3 -\\ x_4 - 4x_5 + 5x_6 +\\ 7x_7 + 3x_8 - 6x_9 +\\ 2x_{10} - 5x_{11} = 7634 \tag{12.5}\end{multline}$$

Another environment for splitting a long equation into multiple lines is **split**. Like the **array** environment, the **split** environment is also nested in another math-mode, such as the **equation** or **displaymath** environment or **\[\]** mode. An application of this environment is shown in Table 12.5. By default the **split** environment makes all the

Table 12.5 Splitting an equation into multiple lines through the **split** environment

LaTeX input	Output

`\begin{equation}`
`  \begin{split}`
`    f(x,y)=h \biggl[ & \frac{1}{2}(x+y)+x^2+y^3\\`
`                     & +\frac{1}{3}z^2\biggr]`
`  \end{split}`
`\end{equation}`
	$$\begin{split} f(x,y)=h \biggl[& \frac{1}{2}(x+y)+x^2+y^3\\ & +\frac{1}{3}z^2\biggr] \end{split} \tag{12.6}$$

lines right aligned. Alignment about another place can be obtained by using an **&** at that place, as shown in Table 12.5 by aligning after the left-hand square bracket. Note that since **\left** and **\right** (used for obtaining auto-sized delimiters) appear as a matching pair, they cannot be split into two lines. In such cases, instead of **\left** and **\right**, the commands of the forms of **\big**, **\Big**, **\bigg** and **\Bigg** (refer §11.1 on page 101 for detail) may be used as shown in Table 12.5, where **\biggl** and **\biggr** are applied for generating two big-sized square brackets in two split lines.

As shown in Table 12.6, the **split** environment can be used in mathematical analysis also, where the multiple lines are aligned about the '=' sign.

Table 12.6 Mathematical analysis through the **split** environment

LaTeX input	Output

`\begin{equation*}`
`  \begin{split}`
`    f(x) &= x^3 + 2x^2 - 5x + 10\\`
`         &= (2)^3 + 2(2)^2 - 5(2) + 10\\`
`         &= 16`
`  \end{split}`
`\end{equation*}`
	$$\begin{split} f(x) &= x^3 + 2x^2 - 5x + 10\\ &= (2)^3 + 2(2)^2 - 5(2) + 10\\ &= 16 \end{split}$$

The \lefteqn{fline} command can also be used in the eqnarray environment for splitting a long equation into multiple lines, where fline is the first split line. The first example in Table 12.7 shows the use of \lefteqn{} for reproducing the

Table 12.7 Splitting an equation into multiple lines through the eqnarray environment

LaTeX input	Output

`\begin{eqnarray}`
`\lefteqn{f(x,y)=h\biggl[\frac{1}{2}(x+y)+x^2+y^3}`
`\nonumber\\`
`&& +\frac{1}{3}z^2\biggr]`
`\end{eqnarray}`
	$f(x,y) = h\biggl[\frac{1}{2}(x+y)+x^2+y^3$
	$+\frac{1}{3}z^2\biggr]$ (12.7)

`\begin{eqnarray}`
`\lefteqn{f(x,y)=h\biggl[\frac{1}{2}(x+y)+x^2+y^3}`
`\hspace{9mm}\nonumber\\`
`&& +\frac{1}{3}z^2\biggr]`
`\end{eqnarray}`
	$f(x,y) = h\biggl[\frac{1}{2}(x+y)+x^2+y^3$
	$+\frac{1}{3}z^2\biggr]$ (12.8)

equation of Table 12.5, where the second line of the equation is preceded by && for printing it with default left indentation. As shown in the second example, the indentation of the following lines can be increased by using \hspace{} after \lefteqn{} in the first part. Since the enqarray environment assigns a default serial number to each equation, \nonumber is used in the first split lines in Table 12.7 for ignoring their numbering.

12.5 Vector and Matrix

The direct LaTeX environments for producing a matrix include matrix, pmatrix, bmatrix, vmatrix and Vmatrix. These environments work for a vector also, where it is to be inserted as a single column or single row matrix. The environments are to be nested in a math-mode, such as the equation environment. Like in the tabular environment used for producing a table (refer §7.1 on page 59), the entries of two columns in a matrix generating environment is separated by a & sign and a row other than the last one is terminated by the line break command \\. The matrix environment produces a matrix without any delimiter, while the delimiters in the pmatrix, bmatrix, vmatrix and Vmatrix environments are (), [ ], | | and ‖ ‖, respectively. Applications of these five environments are shown in Table 12.8 on the next page. Apart from these, there is another environment, smallmatrix, which is useful for producing a small matrix or vector in text-mode, e.g., `$\left[\begin{smallmatrix}a&b\\c&d\end{smallmatrix}\right]$` will produce $\left[\begin{smallmatrix}a&b\\c&d\end{smallmatrix}\right]$.

The above matrix generating environments have some limitations. Firstly, a matrix only up to a maximum of 10 columns can be generated. For a higher order matrix, the value of the column counter MaxMatrixCols is to be reset, e.g., \setcounter{MaxMatrixCols}{15} for producing a matrix up to a maximum of 15

Table 12.8 Matrices and vectors through direct environments

LaTeX input	Output

`\begin{equation*}`
`\begin{matrix} 3a & b \\ c & 5d \end{matrix}`
`\end{equation*}`
	$\begin{matrix} 3a & b \\ c & 5d \end{matrix}$

`\begin{equation*}`
`\begin{pmatrix} x_1 \\ x_2 + 7 \end{pmatrix}`
`\end{equation*}`
	$\begin{pmatrix} x_1 \\ x_2 + 7 \end{pmatrix}$

`\begin{equation*}`
`\begin{bmatrix} 1-y & 0 \\ 0 & 1-y \end{bmatrix}`
`\end{equation*}`
	$\begin{bmatrix} 1-y & 0 \\ 0 & 1-y \end{bmatrix}$

`\begin{equation*}`
`\begin{vmatrix} 50 & 0 \\ 0 & 75 \end{vmatrix}`
`\end{equation*}`
	$\begin{vmatrix} 50 & 0 \\ 0 & 75 \end{vmatrix}$

`\begin{equation*}`
`\begin{Vmatrix} \lambda_1 \\ \lambda_2 + 9 \end{Vmatrix}`
`\end{equation*}`
	$\begin{Vmatrix} \lambda_1 \\ \lambda_2 + 9 \end{Vmatrix}$

columns. Secondly, the environments do not have any column formatting option, but all entries are made center aligned. Moreover, as seen in Table 12.8, no one generates a matrix in { }, which is often used particularly in vectors. However, required delimiters can be produced through the **matrix** environment by enclosing it with the delimiters, e.g., **\left\{\begin{matrix}**...**\end{matrix}\right\}** for producing the matrix in { }.

A good alternative to overcome the limitations of the matrix generating environments is to use the **array** environment. For illustration of the environment, Table 12.9

Table 12.9 Matrices and vectors through the **array** environment

LaTeX input	Output

`\begin{equation*}`
` \vec{\mbox{\boldmath$x$}} = \left\{ \begin{array}{l}`
`x_1\\x_2\\ \vdots\\x_n + k`
`\end{array} \right\}`
`\end{equation*}`
	$\vec{\boldsymbol{x}} = \left\{ \begin{array}{l} x_1\\x_2\\ \vdots\\x_n + k \end{array} \right\}$

`\begin{equation*}`
`\left[ \begin{array}{rrr}`
`33 & 0 & 375\\`
`289 & 470 & 8\\`
`7 & 14 & 67`
`\end{array} \right]`
`\end{equation*}`
	$\left[\begin{array}{rrr} 33 & 0 & 375\\ 289 & 470 & 8\\ 7 & 14 & 67 \end{array} \right]$

shows an example of a column vector in { } with left aligned elements, and a matrix in [] with right aligned elements. Further, as the application of the **array** environment

Table 12.10 Matrix and vector mixed expression through the **array** environment

LaTeX input	Output

`\begin{equation*}`
`\left[\begin{array}{cccc}`
`k_{11} & k_{12} & \ldots & k_{1n}\\`
`k_{21} & k_{22} & \ldots & k_{2n}\\`
`\hdotsfor{4}\\`
`k_{n1} & k_{n2} & \ldots & k_{nn}`
`\end{array}\right]`
`%`
`\left\{\begin{array}{c}`
`x_1\\x_2\\ \hdotsfor{1}\\x_n`
`\end{array}\right\} =`
`%`
`\left\{\begin{array}{l}`
`f_1 + a\\f_2\\ \hdotsfor{1}\\f_n + c`
`\end{array}\right\}`
`\end{equation*}`
	$\left[\begin{array}{cccc} k_{11} & k_{12} & \ldots & k_{1n}\\ k_{21} & k_{22} & \ldots & k_{2n}\\ \hdotsfor{4}\\ k_{n1} & k_{n2} & \ldots & k_{nn} \end{array}\right] \left\{\begin{array}{c} x_1\\x_2\\ \hdotsfor{1}\\x_n \end{array}\right\} = \left\{\begin{array}{l} f_1 + a\\f_2\\ \hdotsfor{1}\\f_n + c \end{array}\right\}$

for generating matrices and vectors, an expression is shown in Table 12.10, which contains a matrix and two vectors. Hence, the expression is inserted in three parts using three individual **array** environments, the first one is for the matrix and the remaining two are for the two vectors. Note that there should not be any line break command or blank line after an **array** environment, otherwise the contents of the next **array** environment will be printed in the following line, instead of in the same line. The `\hdotsfor[aspace]{n}` command draws a horizontal dotted line, in the **array** or any other matrix generating environment, over `n` number of columns with `aspace` as the optional dot spacing, e.g., `\hdotsfor[1.5]{5}` for a line over 5 columns with 1.5 spacing between two dots.

12.6 Overlining and Underlining

Sometime an expression can be presented by putting a line over a term, instead of enclosing it in a pair of delimiters. Such an overlining is done through the `\overline{}` command, an example of which is shown in Table 12.11. Similarly the

Table 12.11 Mathematical expression overlined through the `\overline{}` command

LaTeX input	Output

`\begin{equation*}`
`S = \frac{n}{2}\left(2a+\overline{n-1}d\right)`
`\end{equation*}`
	$S = \frac{n}{2}\left(2a+\overline{n-1}d\right)$

`\underline{}` command can be used for putting a line under a term. On the other hand, the `\overbrace{}` or `\underbrace{}` command can be used for putting a brace over or under a term. Moreover, a note can also be placed over or under such a brace. Examples of both the cases are shown in Table 12.12.

Table 12.12 Mathematical expression with over and under braces

LaTeX input	Output

`\begin{equation*}`
`A = \overbrace{a_{11}+a_{12}+a_{13}}`
`    + \underbrace{a_{21}+a_{22}+a_{23}}`
`\end{equation*}`
	$A = \overbrace{a_{11}+a_{12}+a_{13}} + \underbrace{a_{21}+a_{22}+a_{23}}$

`\begin{equation*}`
`A = \overbrace{a_{11}+a_{12}+a_{13}}^I`
`    + \underbrace{a_{21}+a_{22}+a_{23}}_{II}`
`\end{equation*}`
	$A = \overbrace{a_{11}+a_{12}+a_{13}}^I + \underbrace{a_{21}+a_{22}+a_{23}}_{II}$

A note over an overbrace is put as the superscript to `\overbrace{}`. Similarly, a note under an underbrace is put as the subscript to `\underbrace{}`.

12.7 Stacking Terms*

In many applications, like in chemical reactions, two terms often need to be stacked (to put one above another), where the upper term is usually some texts and lower one is a symbol covering the upper term. Generally stacking is done through the `\stackrel{aup}{alow}`, where `aup` is the upper term and `alow` is the lower one. Some applications of this command are shown in Table 12.13, where arrow symbols are generated through fixed-length based direct arrow commands.

Table 12.13 Stacking a mathematical term with an arrow of fixed length

LaTeX input	Output

`\begin{equation*}`
`  A \stackrel{a}{\rightarrow} B`
`\end{equation*}`
	$A \stackrel{a}{\rightarrow} B$

`\begin{equation*}`
`  \mathrm{2Na + D_2}`
`    \stackrel{\mathrm{heat}}{\longrightarrow}`
`      \mathrm{2NaD}`
`\end{equation*}`
	$\mathrm{2Na + D_2} \stackrel{\mathrm{heat}}{\longrightarrow} \mathrm{2NaD}$

`\begin{equation*}`
`  \mathrm{NH_3}`
`    \stackrel{\mathrm{D_2}}{\rightleftharpoons}`
`      \mathrm{NH_2D}`
`       \stackrel{\mathrm{D_2}}{\rightleftharpoons}`
`          \mathrm{ND_3}`
`\end{equation*}`
	$\mathrm{NH_3} \stackrel{\mathrm{D_2}}{\rightleftharpoons} \mathrm{NH_2D} \stackrel{\mathrm{D_2}}{\rightleftharpoons} \mathrm{ND_3}$

The stacking under `\stackrel{}{}` may look odd if the length of the term is too long or short in comparison to that of the covering symbol. Such problems may arise with arrows whose lengths are predefined. In that situation, stacking may be done through commands like \overleftarrow{}, \overrightarrow{}, \underleftarrow{} and \underrightarrow{}, which produce arrows of flexible lengths to cover their arguments. For proper presentation, \overleftarrow{} and \overrightarrow{}, which produce arrows on top, should be used as subscripts. On the other hand, \underleftarrow{} and \underrightarrow{}, which produce arrows at bottom, should be used as superscripts. Table 12.14 shows applications of \overrightarrow{} and \underrightarrow{} on the

Table 12.14 Stacking a mathematical term with an over or under arrow of flexible length

LaTeX input	Output
\begin{equation*} \mathrm{H_2 + D_2} {}_{\overrightarrow{~\mathrm{600\text{--}750\,^oC}~}} \mathrm{2HD} \end{equation*}	$\mathrm{H_2 + D_2}{}_{\overrightarrow{~\mathrm{600\text{--}750\,^oC}~}}\mathrm{2HD}$
\begin{equation*} \mathrm{H_2 + D_2} {}^{\underrightarrow{~\mathrm{600\text{--}750\,^oC}~}} \mathrm{2HD} \end{equation*}	$\mathrm{H_2 + D_2}{}^{\underrightarrow{~\mathrm{600\text{--}750\,^oC}~}}\mathrm{2HD}$

same example, where the commands are inserted, respectively, as the subscript and superscript to an empty character, i.e., to {} (it can be subscribed to the previous term also). Since \overrightarrow{} and \underrightarrow{} produce arrows of lengths equal to those of their arguments, ~ is added on either side of their arguments for producing arrows of slightly bigger lengths. Further, since the dash producing command '--' does not work in math-mode, it is inserted through the \text{} command.

LaTeX also provides the \overset{atop}{abot} and \underset{abot}{atop} commands for stacking two terms, where atop is produced on the top of abot, e.g., \overset{a}{X}, \underset{b}{X} and \overset{a}{\underset{b}{X}} will produce $\overset{a}{X}$, $\underset{b}{X}$ and $\overset{a}{\underset{b}{X}}$ respectively. Notice the expressions in Table 12.14 – both of \overrightarrow{} and \underrightarrow{} print their arguments on one side of the arrows. If terms on both sides of an arrow are required, either the pair of \underset{}{} and \underrightarrow{}, or \overset{}{} and \overrightarrow{} may be used. Applications of both the pairs are shown in Table 12.15 on the next page, where it is to be noticed that the smaller stacking term is taken as the first argument of \underset{}{} or \overset{}{} (otherwise an arrow of a smaller length will be produced).

The \overset{}{} and \underset{}{} commands can also be used for printing ranges or limits of big symbols like $\sum$ and $\prod$. Generally, the ranges of these symbols are inserted as superscripts and subscripts, and these are printed on the top and at bottom of the symbols, e.g., '\sum^n_{i=1} x_i' in most of the math-modes will produce $\displaystyle\sum_{i=1}^{n} x_i$, but $\sum_{i=1}^n x_i$ in text-mode and array environment (i.e., like superscript and subscript on the right side). In such cases, \overset{}{} and \underset{}{} can be used for forcibly printing the ranges on the top and at bottom of a symbol,

Table 12.15 Stacking two mathematical terms above and below an arrow

LaTeX input	Output
`\begin{equation*} \mathrm{H_2 + D_2} \underset{\mathrm{Ni}} {{}^{\underrightarrow{~\mathrm{600\text{--}750\,^oC}~}}} \mathrm{2HD} \end{equation*}`	$\mathrm{H_2 + D_2} \underset{\mathrm{Ni}}{\xrightarrow{\mathrm{600\text{--}750\,^oC}}} \mathrm{2HD}$
`\begin{equation*} \mathrm{H_2 + D_2} \overset{\mathrm{Ni}} {{}_{\overrightarrow{~\mathrm{600\text{--}750\,^oC}~}}} \mathrm{2HD} \end{equation*}`	$\mathrm{H_2 + D_2} \xrightarrow[\mathrm{600\text{--}750\,^oC}]{\mathrm{Ni}} \mathrm{2HD}$

e.g., '`$\underset{i=1}{\overset{n}{\sum}}$`' or '`$\overset{n}{\underset{i=1}{\sum}}$`' in this line produces $\underset{i=1}{\overset{n}{\sum}}$.

For stacking multiple lines above or below of a symbol, the **\substack{}** command or the **subarray** environment may be used. The lines are center aligned under **\substack{}**, while their alignment can be controlled in the **subarray** environment. Some examples of these options are shown in Table 12.16, where the lines under the **subarray** environment are left aligned through the option **l** (other option is **c** for center alignment).

Table 12.16 Stacking multiple mathematical lines above or below of a symbol

LaTeX input	Output
`\begin{equation*} \sum_{\substack{i=1\\ i\in\Omega_{\text{old}}}} \end{equation*}`	$$\sum_{\substack{i=1\\ i\in\Omega_{\text{old}}}}$$
`\begin{equation*} \prod_{i=1}^{\substack{i=n\\ n=\text{Sl.No}}} \end{equation*}`	$$\prod_{i=1}^{\substack{i=n\\ n=\text{Sl.No}}}$$
`\begin{equation*} \sum_{\begin{subarray}{l} i=1\\ i\in\Omega_{\text{old}} \end{subarray}} \end{equation*}`	$$\sum_{\begin{subarray}{l} i=1\\ i\in\Omega_{\text{old}} \end{subarray}}$$
`\begin{equation*} \prod_{i=1}^{\begin{subarray}{l} i=n\\ n=\text{Sl.No} \end{subarray}} \end{equation*}`	$$\prod_{i=1}^{\begin{subarray}{l} i=n\\ n=\text{Sl.No} \end{subarray}}$$

There is another slightly different command, **\sideset{aleft}{aright}**, which prints `aleft` and `aright`, respectively, on the left and right sides of a symbol, like $\sum$ or $\prod$. Provision is also there for printing four different terms, as superscripts and subscripts, on the four corners of a symbol. For example, `$\sideset{a}{b}\prod$` and `$\sideset{^1_2}{^3_4}\prod$` will print $\sideset{a}{b}\prod$ and $\sideset{^1_2}{^3_4}\prod$ respectively.

12.8 Side-by-Side Equations*

If required for some purpose (say, for comparison), sets of equations can be produced side-by-side along the width of a page. Generally, the **gathered**, **aligned** and **alignedat** environments are used in such cases, nesting in a math-mode, such as the **equation** or **displaymath** environment, or **\[\]** mode. Vertical alignment of the sets of equations can also be made through an optional argument to the environments, whose permissible values are **c** for center alignment, **b** for bottom alignment and **t** for top alignment. These environments are very similar, respectively, with the **gather**, **align** and **alignat** environments discussed in §11.5 on page 107. Two applications of the pair of **aligned** and **gathered** environments are shown in Table 12.17. In the first application, both the sets of equations are center aligned (by default), while in the second application, the first set is bottom aligned and the second set is top aligned.

Table 12.17 Side-by-side equations along the page width

LaTeX input	Output

`\begin{equation*}`
`\begin{aligned}`
` a^2 - b^2 &= (a - b)(a + b)\\`
` (a-b)^2   &= a^2 - 2ab + b^2`
`\end{aligned}\qquad`
`\begin{gathered}`
`(a+b)^2 = a^2 + 2ab + b^2`
`\end{gathered}`
`\end{equation*}`
	$\begin{aligned} a^2 - b^2 &= (a-b)(a+b)\\ (a-b)^2 &= a^2 - 2ab + b^2 \end{aligned}\qquad \begin{gathered} (a+b)^2 = a^2 + 2ab + b^2 \end{gathered}$

`\begin{equation*}`
`\begin{aligned}[b]`
` a^2 - b^2 &= (a - b)(a + b)\\`
` (a-b)^2   &= a^2 - 2ab + b^2`
`\end{aligned}\qquad`
`\begin{gathered}[t]`
`(a+b)^2 = a^2 + 2ab + b^2`
`\end{gathered}`
`\end{equation*}`
	$\begin{aligned}[b] a^2 - b^2 &= (a-b)(a+b)\\ (a-b)^2 &= a^2 - 2ab + b^2 \end{aligned}\qquad \begin{gathered}[t] (a+b)^2 = a^2 + 2ab + b^2 \end{gathered}$