

Tables in L^AT_EX_{2 ϵ} : Packages and Methods

Lapo Filippo Mori

Email mori@northwestern.edu
Website <http://www.lapomori.com/>
Address Department of Mechanical Engineering, Northwestern University, Evanston, IL

Abstract This article aims to provide the tools to correctly create tables in L^AT_EX_{2 ϵ} . This objective is pursued analyzing the typical problems that users find creating tables and possible solutions; particular emphasis is on the packages to be used. Examples are given for each case.

Introduction

Tables are one of the most used objects in scientific publications. Besides the regular L^AT_EX commands, a large number of packages can be used to customize the tables and overcome the limitations that the regular commands have. The documentation on tables is very fragmented because every package modifies just an aspect of the table typeset; hence, authors have to look for information on several guides.

This article explains how to use L^AT_EX_{2 ϵ} and its packages to create tables. Each topic is not studied in depth (the reader can refer to the package manual for this) but the author tries to cover as many aspects as possible highlighting the solutions that he likes most. The choice of the contents comes firstly from the author's experience and then from the many discussions on tables on the forum of G_UI_T (Gruppo Utilizzatori Italiani di T_EX).¹

The text presumes that the reader is familiar with the basics of L^AT_EX_{2 ϵ} , which means that he has read at least one of the many free guides available on the net [8, 21] or a book [1, 5, 10–12, 14–18]. Whenever a package is cited, the article

1. The forum of G_UI_T is at <http://www.guit.sssup.it/forum/>.

Table 1: Table not in agreement of the general typeset rules.

D	P_u	u_u	β	G_f
5 in	269.8 lbs	0.000674 in	1.79	0.04089 psi · in
10 in	421.0 lbs	0.001035 in	3.59	"
20 in	640.2 lbs	0.001565 in	7.18	"

does not give a complete description of how it works, which can be found in the package manual,² but analyzes the main options and suggests their use.

This text is focused on the preparation of tables and does not address problems that are connected to it as for example how to position floating objects or how to modify the caption [3]. For these topics, that are in common with all the other floating objects (e.g. figures), the reader can refer to [20].

1 Basics

1.1 General rules

The typeset of tables should be based on the following rules [7]:

1. never use vertical lines;
2. avoid double lines;
3. place the units in the heading of the table (instead of the body);
4. do not use quotation marks to repeat the content of cells.

To better understand the importance of these simple rules, the reader can compare tab. 1 and 2.

2. The majority of \LaTeX packages has a manual which describes how to use it and usually gives examples. The location of the manual can vary between different distributions but, usually, is under the subfolder `/texmf/doc`.

Table 2: Table in agreement of the general typeset rules.

D (in)	P_u (lbs)	u_u (in)	β	G_f (psi·in)
5	269.8	0.000674	1.79	0.04089
10	421.0	0.001035	3.59	0.04089
20	640.2	0.001565	7.18	0.04089

1.1.1 How to input the code

L^AT_EX does not require the columns (i.e. &) to be aligned, however it is advisable to align them in order to make it easier to modify the code. For example, tab. 3 can be obtained with either this code

```
\begin{table}[tp]%
\caption{Maximum load and nominal tension.}
\label{aggiungi}\centering%
\begin{tabular}{clccc}
\toprule%
D$ & & $P_u$ & & $\sigma_N$ & \\
(in)& & (lbs) & & (psi) & \\
5 & & 285 & & 38.00 & \\
& & 287 & & 38.27 & \\
& & 230 & & 30.67 & \\
10 & & 430 & & 28.67 & \\
& & 433 & & 28.87 & \\
& & 431 & & 28.73 & \\
\end{tabular}
\end{table}
```

or this code

```
\begin{table}[tp]%
\caption{Maximum load and nominal tension.}
\label{aggiungi}\centering%
\begin{tabular}{clccc}
\toprule%
D$&&P_u$&&$\sigma_N$\\
(in)&&(lbs)&&(psi)\\
5&&test 1&&285&&38.00\\
&&test 2&&287&&38.27\\
&&test 3&&230&&30.67\\
10&&test 1&&430&&28.67\\
&&test 2&&433&&28.87\\
&&test 3&&431&&28.73\\
\end{tabular}
\end{table}
```

Table 3: Maximum load and nominal tension.

D (in)		P_u (lbs)	σ_N (psi)
5	test 1	285	38.00
	test 2	287	38.27
	test 3	230	30.67
10	test 1	430	28.67
	test 2	433	28.87
	test 3	431	28.73

1.2 Basic commands

1.2.1 How to create a table

The basic tools that L^AT_EX offers to create tables and matrices are the `tabular`, `tabular*`, and `array` environments. The environment `array` can be used only in math mode; all these three environments generate a minipage. The syntax is

```
\begin{array}[pos]{cols}
rows
\end{array}
```

```
\begin{tabular}[pos]{cols}
rows
\end{tabular}
```

```
\begin{tabular*}{width}[pos]{cols}
rows
\end{tabular*}
```

where the meaning of the arguments is [17]:

`pos` Vertical position. It can assume the following values:

`t` the line at the *top* is aligned with the text baseline;

`b` the line at the *bottom* is aligned with the text baseline;

`none` when no option is specified, the table is centered to the text baseline;

`width` Can only be used with `tabular*` and it defines the total width of the table.

When used, the argument `cols` must contain somewhere `@{\extracolsep{\fill}}`.

`cols` Defines the alignment and the borders of each column. It can have the following values:

- `l` the column is aligned to the *left*;
- `r` the column is aligned to the *right*;
- `c` the column is *centered*;
- `p{wth}` the column is justified and its width is `wth` (the text is inserted into a parbox of width `wth`);
- `*{num}{form}` the format `form` is repeated `num` times; for example `*{3}{|1|}` is equal to `|1|1|1|`.

Besides the format of the columns, we can also specify what can be inserted between two columns with the following symbols:

- `|` draws a vertical line (not recommended);
- `||` draws a double vertical line (not recommended);
- `@{text}` insert the text `text` in every line of the table between the columns where it appears. This command eliminates the space that is automatically inserted between the columns. If some horizontal space is needed between `text` and the columns, it can be inserted with the command `\hspace{}`. The command `\extraspaces\fill` in a `tabular*` environment extends the space between the columns where it appears in order to let the table have the width defined by the user (see for example table 12 which is obtained with the code at page 14). In order to eliminate the space that is automatically inserted between two columns it is possible to use the empty command `@{}`.

`rows` represents the content of the cells of the table for each row that is ended by the command `\\`. In each row the content of each column is separated by the symbol `&`; each row has the same number of cells (i.e. same number of `&`)³ which must be equal to that declared in the definition `cols`.

`\hline` can be placed in the first row or at the end of a row `\\` and it draws an horizontal line as wide as the entire table.

`\cline{n-m}` draws an horizontal line from the left of column `n` up to the right of the column `m`.

3. The cells might be empty.

Table 4: Example of the standard L^AT_EX multicolumn and cline commands.

Sample	Roughness R_a (nm)
A ring	385
plate	397
B ring	376
plate	390

`\multicolumn{num}{col}{text}` combines the following num columns into a single cell that has the same width, including the spaces between the columns. The argument col must contain a position symbol l, r o c.

An example of these commands is reported in table 4 which is generated by the following code:

```
\begin{tabular}{llc}
\hline%
\multicolumn{2}{c}{Sample} & Roughness  $R_a$ \\
& & (nm)\hline%
A & ring & 385\\
& plate & 397\hline
B & ring & 376\\
& plate & 390\hline
\end{tabular}
```

Since the command `@{...}` inserts its argument without any space between the columns, it can be used without any argument if we want to eliminate that space. For example in some cases it is useful to eliminate the space on the left of the first column and on the right of the last column; table 5 is obtained with the following code:

```
\begin{tabular}{@{}>\bfseries}lp{6cm}@{}}
...
\end{tabular}
```

Compare the result with table 9.

1.2.2 Parameters that control the style of a table

There are some parameters that control the style of a table and to which L^AT_EX assigns a default value. These commands can be modified by the user either

Table 5: Table without horizontal space on the left of the first column and on the right of the last one.

Force	Force is a vector quantity defined as the rate of change of the momentum of the body that would be induced by that force acting alone.
Moment of a force	Moment of a force with respect to an origin is defined as the cross product of the position vector (with respect to the same origin) and the force.

globally in the preamble or locally inside an environment.

`\tabcolsep` is half of the width of the space between the columns of the `tabular` and `tabular*` environments.

`\arraycolsep` is half of the width of the space between the columns of the `array` environments.

`\doublerulesep` is the space between double lines (`\hline\hline`).

The command `\setlength` can be used to modify these parameters [1, 5, 8, 10, 11, 14, 15, 17, 18, 21].

1.2.3 Floating tables

Usually the tables are used as floating objects [7, 17, 18, 20]. In this case the `tabular` environment must be inserted in a `table` environment which:

- allows to generate the list of tables with the command `\listoftables` (similar to `\listoffigures` for the figures).
- allows to create a caption taking care of:
 - assigning the right name to the environment; this depends on the language selected with `babel`: in Italian it is “*tabella*” while in English is “*table*”;

- assigning a number to the table.
- allows to assign a `\label` to the table which can be used for references in the text.

To center a floating table,⁴ the `\centering` command must be used instead of the `center` environment because the latter inserts an undesired additional vertical space [6, 22].

For example table 5 is obtained with the following code:

```
\begin{table}[tp]
\caption{...} \label{...} \centering\small
\begin{tabular}{...}
...
\end{tabular}
\end{table}
```

1.3 Multirow cells

As `\multicolumn` allows to have cells on more than one column, the `\multirow` command allows to have cells on more than one row. This command requires the package `multirow`.

`\multirow` can be used in two different ways:

`\multirow{row}*{text}` creates a cell that contains the text `text` and extends on row `rows` and has an undefined width;

`\multirow{row}{larg}*{test}` creates a cell that contains the text `text` and extends on row `rows` and has an width equal to `larg`;

Table 6 is an example of a table with `multirow` cells and is generated by the following code:

```
\begin{tabular}{c|cc}
\toprule%
\multicolumn{2}{c}{D} & P_u & \sigma_N \\
\multicolumn{2}{c}{(in)} & (lbs) & (psi) \\
\multirow{3}{*}{5} & test 1 & 285 & \\
38.00 \\ \cmidrule(1){2-4}
& test 2 & 287 & 38.27 \\ \cmidrule(1){2-4}
& test 3 & 230 & 30.67 \\ \midrule
\multirow{3}{*}{10} & test 1 & 430 & \\
28.67 \\ \cmidrule(1){2-4}
```

4. Note that this is valid for every floating object.

Table 6: Example of the standard L^AT_EX multicolumn and multirow commands.

	D (in)	P_u (lbs)	σ_N (psi)
5	test 1	285	38.00
	test 2	287	38.27
	test 3	230	30.67
10	test 1	430	28.67
	test 2	433	28.87
	test 3	431	28.73

```

& test 2          & 433          & 28.87\\cmidrule(1){2-4}
& test 3          & 431          & 28.73\\bottomrule
\end{tabular}

```

Compare it with table 3 that does not use the `\multirow` command.

1.4 Aesthetic improvements

The tabular environment provided by L^AT_EX gives a not satisfactory typographic result because the vertical space between the horizontal lines (obtained with `\hline`) and the text is too thin.

To solve this problem, the `booktabs` and `ctable` packages offer the `\toprule`, `\midrule`, and `\bottomrule` commands to be used instead of `\hline`. In particular `\toprule` and `\bottomrule` must be used for the first and the last line and are thicker than the others, obtained with `\midrule`. `\cmidrule` is to be used instead of `\cline`. These commands modify the thickness of the lines and the vertical space before and after them; in particular `\toprule` and `\bottomrule` are thick and with a wider space below and above respectively, while `\midrule` is thin and with above and below space of the same width. Compare tables 2 and 7.

The default thickness is 0.08 em for `\toprule` and `\bottomrule` and 0.05 em for `\midrule`. To modify the thickness of a line locally (i.e. on a single line) the user can insert the thickness in square brackets after the line command; for example

Table 7: Table obtained with the standard L^AT_EX lines (`hline`).

D (in)	P_u (lbs)	u_u (in)	β	G_f (psi · in)
5	269.8	0.000674	1.79	0.04089
10	421.0	0.001035	3.59	0.04089
20	640.2	0.001565	7.18	0.04089

the command

```
\midrule[0.08em]
```

modifies locally the thickness of a `\midrule`. To modify the thickness of lines globally (i.e. in the whole document), the user can modify the `\heavyrulewidth` (which controls the thick lines, i.e. `\toprule` and `\bottomrule`) and `\lightrulewidth` (which controls the thin lines, i.e. `\midrule`) lengths. For example in this article the thick lines are 0.1 em thick due to the command

```
\setlength{\heavyrulewidth}{0.1em}
```

In this article the line below the heading of the tables has always thickness equal to `\toprule` and `\bottomrule` but it is vertically centered with respect to the row above and below (which is typical of the lines `\midrule`). To obtain this result, a new line type has been defined with the following command

```
\newcommand{\otoprule}{\midrule[\heavyrulewidth]}
```

2 How to align the text inside a cell

L^AT_EX provides four types of alignment (see paragraph 1.2.1): left alignment (`l`), right alignment (`r`), centering (`c`) and justified with assigned width (`p{wth}`). Specific packages provide more types of horizontal alignment for the text.

2.1 The array package

The array package defines new options for column alignment in the array and tabular environments:

`m{wth}` defines a justified column with width `wth` and cells that are vertically centered (as a `\parbox[c]{wth}`);

`b{wth}` defines a justified column with width `wth` and cells that are aligned to the bottom (as a `\parbox[b]{wth}`);

`>{ins}` can be placed before a command `l`, `r`, `c`, `p`, `m` or `b` and inserts `ins` before the content of the cell;

`<{ins}` can be placed after a command `l`, `r`, `c`, `p`, `m` or `b` and inserts `ins` after the content of the cell.

If the content of the table is only mathematics, it can be handy to define it in the preamble of the table instead of using `$. . .$` in each cell. For example it is possible to define new types of column

```
\newcolumntype{A}{>{$}c<{$}}
\newcolumntype{Q}{>{$}l<{$}}
\newcolumntype{V}{>{$}r<{$}}
```

to have column respectively centered, aligned to the left or to the right. If the cell have to contain mathematics in the `displaystyle` format we can define the columns as

```
\newcolumntype{A}{>{$\displaystyle}c<{$}}
\newcolumntype{Q}{>{$\displaystyle}l<{$}}
\newcolumntype{V}{>{$\displaystyle}r<{$}}
```

For example the code

```
\newcolumntype{Q}{>{$\displaystyle}l<{$}}
\newcolumntype{A}{>{$}c<{$}}
\begin{tabular}{QA}
\toprule%
\int \cos x, dx & & \sin x + c & \\ \midrule
\int e^x dx & & e^x + c & \\ \midrule
\int \sec^2 x, dx & & \tan x + c & \\ \bottomrule
\end{tabular}
\end{table}
```

gives table 8.

Further details on the array package can be found on its manual and [13].

2.1.1 How to format the text of a column

The commands `>{ins}` and `<{ins}` can also be used to format certain columns: it is possible to use L^AT_EX commands `\upshape`, `\itshape`, `\slshape`, `\scshape`, `\mdseries`, `\bfseries`, `\rmfamily`, `\sffamily`, and `\ttfamily` [18].

For example the code

Table 8: Table with mathematics obtained with the array package.

$\int \cos x dx$	$\sin x + c$
$\int e^x dx$	$e^x + c$
$\int \sec^2 x dx$	$\tan x + c$

Table 9: Table with automatic boldface text on one column (array package).

Force	Force is a vector quantity defined as the rate of change of the momentum of the body that would be induced by that force acting alone.
Moment of a force	Moment of a force with respect to an origin is defined as the cross product of the position vector (with respect to the same origin) and the force.

```

\begin{tabular}{>{\bfseries}l p{6cm}}
\toprule
Force & Force is a vector quantity defined as the rate of
change of the momentum of the body that would be induced by that
force acting alone.\\ \midrule
Moment of a force & Moment of a force
with respect to an origin is defined as the cross product of the
position vector (with respect to the same origin) and the
force.\\ \bottomrule
\end{tabular}

```

produces table 9 where the text in the first column is boldface.

Table 10: Use of the array package to format the text of a row.

Quantity	Symbol	Unit	Value
Stiffness in z direction	k_z	N/m	2276
Stiffness in r direction	k_r	N/m	3414
Weight of the body	P	N	35

2.1.2 How to format the text of a row

There are no commands or packages to format rows but it is possible to define a new command [6] which uses the tools of the array package discussed above. The following definitions are added to the preamble

```
\newcolumntype{+}{>{\global\let\currentrowstyle\relax}}
\newcolumntype{^}{>{\currentrowstyle}}
\newcommand{\rowstyle}[1]{\gdef\currentrowstyle{#1}%
#1\ignorespaces
}
```

and then + must be added on the left of the first column and ^ on the right of every other column in the column definition of tabular. This defines a new command

```
\rowstyle{...}
```

which can be inserted at the beginning of a row to format its cells. For example the code

```
\begin{tabular}{+l^c^c}
\toprule\rowstyle{\bfseries}%
Quantity & Symbol & Unit & Value \\ \otoprule%
...
\end{tabular}
```

produces table 10 where the first row is boldface.

2.2 The tabularx package

The tabularx package requires the same arguments of tabular* but, in order to let the table have the width specified by the user, it modifies the width of certain columns instead of the space between columns. The columns that can be stretched are identified by the alignment command X. This package requires the array package. For example the commands

Table 11: Table obtained with the tabularx package.

Force	Force is a vector quantity defined as the rate of change of the momentum of the body that would be induced by that force acting alone.
Moment of a force	Moment of a force with respect to an origin is defined as the cross product of the position vector (with respect to the same origin) and the force.

```
\begin{tabularx}{\textwidth}{>{\bfseries}lX}
\toprule
Force & Force is a vector quantity defined as the rate of change of
the momentum of the body that would be induced by that
force acting alone.\\
\midrule
Moment of a force & Moment of a force with respect to an origin is
defined as the cross product of the position vector (with respect to
the same origin) and the
force.\\
\bottomrule
\end{tabularx}
```

produce table 11; notice that in the similar table 9 the length of the second column has been defined by the user, while in table 11 the user defines the total width of the table (in this case `\textwidth`) and then the program computes automatically the width of the second column. Compare the result with table 12 obtained with the `tabular*` environment: in this case the space between the columns is stretched instead of the width of the column. To obtain the table, the following code was used

```
\begin{tabular*}{\textwidth}[tb]{%
  >{\bfseries}l@{\extracolsep{\fill}}p{6cm}}
...
\end{tabular*}
```

The X column are by default of type `p{wth}` (i.e. justified) and the width `wth` is determined automatically by the program. It is anyway possible to define new column types with a different horizontal alignment with the command `\newcolumntype`. For example

```
\newcolumntype{Y}{>{\raggedright\arraybackslash}X}
```

defines columns aligned to the left,

```
\newcolumntype{W}{>{\raggedleft\arraybackslash}X}
```

Table 12: Table obtained with the `tabular*` environment.

Force	Force is a vector quantity defined as the rate of change of the momentum of the body that would be induced by that force acting alone.
Moment of a force	Moment of a force with respect to an origin is defined as the cross product of the position vector (with respect to the same origin) and the force.

defines columns aligned to the right,

```
\newcolumntype{Z}{>{\centering\arraybackslash}X}
```

defines centered columns. For example the code

```
\begin{tabularx}{\textwidth}{YXZW}
\toprule
Cell with text aligned to the left & 1 & 2 & 3\\midrule
4 & Cell with justified text & 5 & 6\\midrule
7 & 8 & Cell with centered text & 9\\midrule
10 & 11 & 12 & Cell with text aligned to the right\\bottomrule
\end{tabularx}
```

produces table 13 which is as wide as the text (`\textwidth`).

The X column are by default of type `p{wth}` corresponding to a `\parbox[t]` which means that vertically the cells are aligned to the top. It is anyway possible to define new column types with a different vertical alignment with the command `\tabularxcolumn`. For example

```
\renewcommand{\tabularxcolumn}[1]{>\arraybackslash}m{#1}
```

defines columns vertically aligned to the center,

```
\renewcommand{\tabularxcolumn}[1]{>\arraybackslash}b{#1}
```

defines columns vertically aligned to the bottom. The code

```
\renewcommand{\tabularxcolumn}[1]{>\arraybackslash}m{#1}
\begin{tabularx}{\textwidth}{YXZW}
\toprule Cell with text aligned to the left & 1 & 2 & 3\\midrule 4 &
Cell with justified text & 5 & 6\\midrule 7 & 8 & Cell with centered
```

Table 13: Table obtained with the tabularx package and redefining new types of horizontal alignment for the columns.

Cell with text aligned to the left	1	2	3
4	Cell with justified text	5	6
7	8	Cell with centered text	9
10	11	12	Cell with text aligned to the right

```
text & 9\\midrule 10 & 11 & 12 & Cell with text aligned to the
right\\bottomrule
\\end{tabularx}
```

produces table 14(a), while the same code with the command

```
\\renewcommand{\\tabularxcolumn}[1]{>{\\arraybackslash}b{#1}}
```

produces table 14(b).

The tabularx package is also useful when some columns must have the same width but it is not required to define it directly. In fact, if several X columns are used in the same table, their width is the same. For example table 15 is obtained with the following code

```
\\newcolumnstype{K}{>{\\centering\\arraybackslash}X<{}}
\\begin{tabularx}{.7\\textwidth}{*{7}{K}}
...
\\end{tabularx}
```

By default the width of the columns is automatically computed by the program, anyway tabularx allows to assign the width of some columns. This can be done only if the sum of the widths X remains the same. If for example a column is supposed to be half as wide as another one and the table has only two X columns (these two), the first one must be $\frac{2}{3} \cdot x$ wide and the second one $\frac{4}{3} \cdot x$; in this way they are in the right proportion and their sum is equal to $2x$ (which is the width of two X columns); to obtain this result the following command can be used

Table 14: Tables obtained with the `tabularx` package and defining the vertical alignment of the cells in order to have them centered (a) and aligned to the bottom (b).

(a) Vertically centered cells

Cell with text aligned to the left	1	2	3
4	Cell with justified text	5	6
7	8	Cell with centered text	9
10	11	12	Cell with text aligned to the right

(b) Cells aligned to the bottom

Cell with text aligned to the left	1	2	3
4	Cell with justified text	5	6
7	8	Cell with centered text	9
10	11	12	Cell with text aligned to the right

Table 15: Table obtained with the `tabularx` package to have columns with the same width.

α	$\sin \alpha$	$\cos \alpha$	$\tan \alpha$	$\csc \alpha$	$\sec \alpha$	$\cot \alpha$
-2π	0	1	0	∞	1	∞
$-7\pi/4$	$\sqrt{2}/2$	$\sqrt{2}/2$	1	$\sqrt{2}$	$\sqrt{2}$	1
$-3\pi/2$	1	0	∞	1	∞	0
$-5\pi/4$	$\sqrt{2}/2$	$-\sqrt{2}/2$	-1	$\sqrt{2}$	$-\sqrt{2}$	-1
$-\pi$	0	-1	0	∞	-1	∞
$-3\pi/4$	$-\sqrt{2}/2$	$-\sqrt{2}/2$	1	$-\sqrt{2}$	$-\sqrt{2}$	1
$-\pi/2$	-1	0	∞	-1	∞	0
$-\pi/4$	$-\sqrt{2}/2$	$\sqrt{2}/2$	-1	$-\sqrt{2}$	$\sqrt{2}$	-1
0	0	1	0	∞	1	∞

Table 16: Table obtained with the `tabularx` package imposing the proportion on the width of columns: the first column is half as wide as the second one.

Force	Force is a vector quantity defined as the rate of change of the momentum of the body that would be induced by that force acting alone.
Moment of a force	Moment of a force with respect to an origin is defined as the cross product of the position vector (with respect to the same origin) and the force.

```
<{\hspace=0.66\hspace}X>{\hspace=1.34\hspace}X
```

Table 16 is obtained adding this line to the code of table 11.

2.3 The tabulary package

The `tabulary` package provides results similar to those obtained with the `tabularx` package in table 13. Instead of redefining the alignment of the column with the command `\newcolumntype` as shown in par. 2.2, the `tabulary` package provides the commands R (aligned to the right), C (centered), L (aligned to the left) e J (justified).

For example table 13 can be obtained also with the code

```
\begin{tabulary}{\textwidth}{LJCR}
\toprule
Cell with text aligned to the left & 1 & 2 & 3\\midrule
4 & Cell with justified text & 5 & 6\\midrule
7 & 8 & Cell with centered text & 9\\midrule
10 & 11 & 12 & Cell with text aligned to the right\\bottomrule
\end{tabulary}
```

2.4 How to break lines manually

If the content of a cell is too long, it is necessary to break it into more lines. The packages presented above (`tabularx` and `tabulary`) solve this problem automatically and then they are preferred. In the cases in which it is not possible to use them it is still possible to break the lines manually. If, for example, we want to break

Table 17: Table with a column of defined width and alignment to the right.

Force	Force is a vector quantity defined as the rate of change of the momentum of the body that would be induced by that force acting alone.
Moment of a force	Moment of a force with respect to an origin is defined as the cross product of the position vector (with respect to the same origin) and the force.

the lines only in certain cells of a column, it is possible to insert into these cells a parbox or a minipage.

The use of these commands is also convenient when we want to assign the width of a non justified column.⁵ If for example we want a column 5 cm wide and with text aligned to the right, it is possible to insert all the cells of that column into minipage environments with the command

```
<{\bfseries}l>{\begin{minipage}[t]{5cm}%
\raggedleft\arraybackslash}%
l<{\end{minipage}\arraybackslash}}
```

With this command it is possible to obtain table 17 with the code used for table 11.

2.5 How to change the horizontal alignment of certain cells

To modify the alignment of certain cells, it is possible to insert the text into a *box* (for example with the command `\makebox`). For example table 18 is obtained with the following code:

```
\begin{tabular}{lp{3cm}}
\hline
Text & to the left\\\hline
```

5. If the column has to be justified we can just use a `p{wth}`.

Table 18: Table with cell with different alignment in the same column.

Text	to the left
Text	at the center
Text	to the right

```
Text & \makebox[3cm][c]{at the center}\\\hline
Text & \makebox[3cm][r]{to the right}\\\hline
\end{tabular}
```

2.6 How to align the numbers to the decimal point

2.6.1 Standard commands

The command `@{.}` can be used to align the numbers to the decimal point if the integral part of the number is on one column and the fractional part on the next one. For example table 20 can be obtained with the following code:

```
\begin{tabular}{c r @{.} l}
\toprule
Expression & \multicolumn{2}{c}{Value} & \\
\midrule
 $\pi$  & 3 & 1416 & \\
 $\pi^{\pi}$  & 36 & 46 & \\
 $\pi^{\pi^{\pi}}$  & 80662 & 7 & \\
\bottomrule
\end{tabular}
```

2.6.2 The dcolumn package

If we do not want to break the fractional and the integral part in two columns, the dcolumn package provides a new type of column

```
D{sep-in}{sep-out}{before.after}
```

The first argument (`sep-in`) is the symbol used in the `.tex` document to separate the integral and the fractional part (usually the decimal point `.` or the decimal comma `,`), the second argument (`sep-out`) is the symbol that we want in the output, the third is the number of digits on the left (`before`) and on the right (`after`) of this symbol. The numbers are aligned to the decimal point and, in case that the third argument is negative, the decimal point is aligned to the center of the column. If the columns have a heading, it must be inserted into the command

Table 19: Table with alignment to the decimal point.

Expression	Value
π	3.1416
π^π	36.46
π^{π^π}	80662.7

`\multicolumn{1}{c}{...}`. For example table 20 can be obtained also with the following code

```
\begin{tabular}{cD{,}{.}{5.4}}
\toprule
Espressione & \multicolumn{1}{c}{Valore} \\ \otoprule
$\pi$ & 3,1416 \\ \midrule
$\pi^\pi$ & 36,46 \\ \midrule
$\pi^{\pi^\pi}$ & 80662,7 \\ \bottomrule
\end{tabular}
```

In order to avoid inserting all the three parameters for every column, it is possible to define a new type of column, for example with the code

```
\newcolumntype{d}[1]{D{,}{.}{#1}}
```

where `d` has only one argument which defines the number of decimal digits.

Gregorio [13] shows an interesting use of `\newcolumntype` to let the decimal separator be dependent on the language selected with `babel`: comma (,) for the italian option and point (.) for the english option.

2.6.3 The `rccol` package

`rccol` is a more advanced package that allows to align numbers to the decimal point. This package has the same capabilities of `dcolumn` but in addition can round the numbers and even add zeros in order to have all the cells with the same number of decimal digits. The columns to be aligned to the point are defined with the code

```
R[sep-in][sep-out]{before}{after}
```

where the arguments have the same meaning of those for `dcolumn`. The argument `after` in this case defines the number of digits to keep during the approximation; for example if this parameter is equal to 3, 3.1416 becomes 3.142 and 80662.7

Table 20: Table with alignment to the decimal point and automatic approximation.

Expression	Value
π	3.142
π^π	36.460
π^{π^π}	80662.700

becomes 80662.700. These parameters assign to the number a precision of 10^{-n} (where n is the value of after) and thus it can also assume negative values; for example if this parameter is equal to -1 , 3.1416 becomes 0 and 80662.7 becomes 80660.

For example the code

```
\begin{tabular}{cR[,][.]{5}{3}}
\toprule
Expression          & \multicolumn{1}{c}{Value} & \\ \otoprule
$\pi$                & 3,1416                    & \\ \midrule
$\pi^{\pi}$         & 36,46                     & \\ \midrule
$\pi^{\pi^{\pi}}$   & 80662,7                   & \\ \bottomrule
\end{tabular}
```

produces table 20.

3 Large tables

Sometimes, especially in the appendix, the size of a table is bigger than a page in width, height, or both. In case the table is too long, it is possible to

- reduce the size of the font (see par. 3.1);
- scale it down (see par. 3.3);
- break it over several pages (see par. 3.4).

In case the table is too wide, it is possible to

- reduce the size of the font (see par. 3.1);
- rotate it (see par. 3.2);

Table 21: Table with font size smaller (`footnotesize`) than the rest of the text (`normalsize`).

Force	Force is a vector quantity defined as the rate of change of the momentum of the body that would be induced by that force acting alone.
Moment of a force	Moment of a force with respect to an origin is defined as the cross product of the position vector (with respect to the same origin) and the force.

- scale it down (see par. 3.3);
- break it over several pages (see par. 3.4).

Obviously in each case it is possible to use more solutions at once.

3.1 How to reduce the size of the font

Reducing the size of the font in a table just requires the font size command to be inside the table environment (or similar). For example the code

```
\begin{table}[tp]\footnotesize
\caption{...} \label{...} \centering
\begin{tabular}{...}
...
\end{tabular}
\end{table}
```

produces table 21; compare the result with table 9.

In case the table is non floating (i.e. it is not inside the table environment or similar) the font size command must be outside the tabular environment and then, at the end of the table, the font size used in the rest of the document must be reestablished. On the contrary, when the font size command is inside a table environment (or similar) it affects just the text inside this environment and it is not necessary to reestablish the normal font size at the end.

3.2 Rotating tables

The methods to rotate tables are also valid for every other floating object; [23] is an excellent reference on this topic.

3.2.1 The graphicx package

The graphicx package provides the command

```
\rotatebox[options]{angle}{object}
```

that allows to rotate every object (or box) and can also be used for tables; `options` is an optional argument to modify the origin of the rotation (see [23]), `angle` is the angle of the rotation (for tables is typically 90°) and `object` is the object to be rotated (in this case tabular).

3.2.2 The rotating package

The rotating package provides commands that rotate specifically floating objects and can be used also with tables. The first thing to do is to define the positive direction for rotations with the commands

`clockwise` (default) measures the angle clockwise to be compatible with [10], where the positive angles are always clockwise.

`counterclockwise` as the previous but with positive angles counterclockwise.

The rotating package provides the `rotate`, `turn`, `sideways`, and `sidewaystable` environments.⁶

The rotate environment. This environment does not leave vertical space for the rotated object thus it cannot be used for tables because they would overwrite the preceding text.

```
\begin{rotate}{angle}  
...  
\end{rotate}
```

6. The package also provides `sidewaysfigure` which is the analogous of `sidewaystable` for figures.

The turn environment. This environment, unlike `rotate`, adds a vertical space for the rotated object and thus behaves like `\rotatebox` (see par. 3.2.1).

```
\begin{turn}{<degree>}
...
\end{turn}
```

The sideways environment. This environment is a particular turn environment: it rotates the object by 90° and leaves vertical space for it; in this case the user does not have to indicate the angle. Since wide tables must be rotated by 90° , this is the most suitable environment.

```
\begin{sideways}
\begin{tabular}
...
\end{tabular}
\end{sideways}
```

The sidewaysstable environment. This environment does the same rotation sideways does but it must be used with floating tables for which it substitutes the `table` environment. The syntax is

```
\begin{sidewaystable}
\caption{...}
\begin{tabular}
...
\end{tabular}
\end{sidewaystable}
```

Unlike the preceding environments, `sidewaystable` uses a whole page. Since it is a floating environment, if necessary, L^AT_EX fills the preceding page with the text that comes after the table.

For example the code

```
\begin{sidewaystable}[p]\small
\caption{...} \label{...} \centering
\renewcommand{\tabularxcolumn}[1]{>{\arraybackslash}m{#1}}
\newcolumntype{W}{>{\centering\arraybackslash}X}
\begin{tabularx}{\textheight}{lccWWWWW}
...
\end{tabularx}
\end{sidewaystable}
```

produces the table at page 1; in this example, besides `sidewaystable`, also the `tabularx` package is used to define a `W` type column (horizontally centered text),

Table 1: Comparison of performance improvement of blast-resistant structures.

Quantity	Symbol	Unit	Square honeycomb sandwich			Pyramidal honeycomb sandwich
			i	ii	iii	
Core relative density	ρ_c	%	4.0	4.0	4.0	4.1
Mass per area	m_A	kg/m ²	14.0	14.0	14.0	13.78
Experimental results	\dot{I}	—	1.234	0.931	0.939	0.925
	δ_{max}/L	—	0.391	0.299	0.297	0.299
Given Impulse	I_N	%	28.9	27.6	28.7	27.2
	I_L	%	1.240	1.217	1.296	1.210
Given deflection	I_N	%	40.6	38.0	40.2	37.1
	I_L	%	—	—	—	—

Figure 1: Example of the sideways stable environment.

the text is vertically centered, and the font size is reduced with the command `\small`.

3.2.3 The lscape package

The lscape package defines the landscape environment which rotates its content by 90°. This environment has been designed to work with the longtable package and so its content can even span over several pages.

3.3 Scaling tables

The `graphicx` package provides two commands that can be used to scale tables: `\scalebox` and `\resizebox`.

The command

```
\scalebox{h-scale}[v-scale]{argument}
```

scales `argument` (in this case a table) by `h-scale` horizontally and `v-scale` vertically; for tables the same scale factor must be used horizontally and vertically and this can be done assigning one parameter between `h-scale` and `v-scale` and assigning `!` to the other one.

The command

```
\resizebox{width}{height}{argument}
```

scales `argument` (in this case a table) in order to have a width equal to `width` and a height equal to `height`; in this case is also possible to use `!` to maintain the same aspect ratio of the table. Typically these commands are used to scale down large tables in order to let them be as wide as the text; this can be done for example with the commands

```
\begin{table}[tb]
\caption{...} \label{...} \centering
\resizebox{\textwidth}{!}{%
\begin{tabular}{...}
...
\end{tabular}}
\end{table}
```

For example tables [14\(a\)](#) and [14\(b\)](#) are obtained with the same code of [13](#) but are scaled with `\resizebox` so that their width is 65% of that of the text:

```
\resizebox{0.65\textwidth}{!}{...}
```

3.4 Tables over several pages

The `tabular` environment must always be contained in a single page: if bigger, the parts that exceed the page boundaries are cut and an `Overfull vbox` error message is displayed. Several packages can be used to overcome this limitation and break tables over several pages.

3.4.1 The supertabular package

The supertabular package provides the homonymic supertabular environment that behaves like the regular tabular but, when compiling, it checks the length of the table at every row: when the length exceeds `textheight` the package automatically inserts the optional argument `tabletail` and the `\end{tabular}` command; then it starts a new table on the next page, after the optional argument `tablehead`. The `tabletail` argument can be used to place at the bottom of every page in which the table is broken “continues on the next page” while `tablehead` for “continues from the previous page”. The supertabular package treats the part of the table that is in the new page independently thus the width of the column can vary across different pages unless the user uses fixed width columns.

3.4.2 The xtab package

The xtab package has functions very similar to those of supertabular but it corrects some defects thus it is recommended. It also allows to have a different caption for the last page (`\tablelasthead`).

3.4.3 The longtable package

The longtable package builds pieces of the table during the first compilation and then uses the information that it wrote in the `.aux` file to decide where to break the table during the following compilations. In this way the columns have the same width in every page. The package requires several compilation (as opposed to supertabular) and it is incompatible with many other packages.

3.4.4 The lscape package

The lscape package is meant to rotate by 90° the supertabular environment for wide and long tables.

The reader is suggested to read the manual for all the packages reported above since they report several examples and describe all the options. As regards longtable, [13] is also suggested.

4 Tables and colors

A colored background can be very useful when trying to highlight a part of a table. The `colortbl` package can be used to color the background of cells, rows or columns of `tabular` environments; it can also be used to color the lines (for example `\hline`) but this topic is not analyzed since scientific document should never use colored lines. The `colortbl` requires the `color` and `array` packages. The `xcolor` [4] package can be used instead of `color` but the examples reported below use only the latter.

4.1 How to color columns

The package provides the command `\columncolor` that must be used only inside the command `>{...}` (see par. 2.1). The syntax is

```
\columncolor[clrmodel]{color}[left overhang][right overhang]
```

The first two arguments are standard commands of the `color` package [2], used as in the command `\color`, while the last two arguments are lengths.

`clrmodel` declares the color model to be used; `rgb`, `cmymk`, `gray`, and `named` are available.

`color` declares the color to be used; this definition depends on the color model chosen. `rgb` (Red Green Blue) requires a list of three numbers between 0 and 1 and separated by commas; each of them gives the corresponding color component (red, green, and blue). `cmymk` (Cyan Magenta Yellow Black) requires a list of four numbers between 0 and 1 and separated by commas each of them gives the corresponding color component (cyan, magenta, yellow, and black). `gray` requires just a number between 0 and 1 of the corresponding gray level; in scientific tables grays are preferred to colors thus this command is very convenient. `named` allows to use colors based on their name; the name, if not already defined by default, must be defined by the user with the command

```
\definecolor{name}{clrmodel}{color}
```

where `name` is the color name assigned by the user, `clrmodel` and `color` have the same meaning they have in the command `\columncolor`.

Table 22: Table with a column with colored background obtained with the colortbl package.

D (in)	P_u (lbs)	u_u (in)	β	G_f (psi·in)
5	269.8	0.000674	1.79	0.04089
10	421.0	0.001035	3.59	0.04089
20	640.2	0.001565	7.18	0.04089

`left overhang` and `right overhang` specifies how much the colored background is separated from the text on the left and on the right respectively. The use of these commands is not recommended since, if they are not equal to zero, the colored background does not fill all the space reserved to `\tabcolsep`.

The `\newcolumntype` command (see par. 2.1) can be used to redefine new column types with the desired background color. For example the command

```
\newcolumntype{K}{\columncolor[gray]{0.8}\raggedright}
```

allows to obtain table 23 with the code

```
\begin{tabular}{ccccU}
\hline%
$D$ & $P_u$ & $u_u$ & $\beta$ & $G_f$ \\
(in) & (lbs) & (in) & & (\text{psi}\cdot\text{in}) \\
5 & 269.8 & 0.000674 & 1.79 & 0.04089 \\
10 & 421.0 & 0.001035 & 3.59 & 0.04089 \\
20 & 640.2 & 0.001565 & 7.18 & 0.04089 \\
\end{tabular}
```

Compare the result with table 7. Note that the commands provided by the `ctable` package (see par. 1.4) cannot be used for the horizontal lines: these commands leave vertical space above and below every cell and so the background color would be interrupted; see for example table 22.

4.2 How to color rows

The `\rowcolor` command can be used to color rows: it has the same syntax of `\columncolor` and must be placed at the beginning of the row. If both `\columncolor` and `\rowcolor` are used in a table, the latter prevails. For example table 24 is obtained with the same code of table 2 just adding the command

```
\rowcolor[gray]{.8}
```

Table 23: Table with a column with colored background (ctable package).

D (in)	P_u (lbs)	u_u (in)	β	G_f (psi·in)
5	269.8	0.000674	1.79	0.04089
10	421.0	0.001035	3.59	0.04089
20	640.2	0.001565	7.18	0.04089

Table 24: Table with a row with colored background (\rowcolor command).

D (in)	P_u (lbs)	u_u (in)	β	G_f (psi·in)
5	269.8	0.000674	1.79	0.04089
10	421.0	0.001035	3.59	0.04089
20	640.2	0.001565	7.18	0.04089

Sometimes the background color is not used to highlight certain rows but to separate them (as horizontal lines do). In this case the xcolor package is very handy because it provides commands to automatically color all the odd and even lines of the table. The table option of the package must be loaded in the preamble

```
\usepackage[table]{xcolor}
```

and then the following command must be used right before the tabular environment

```
\rowcolors{row}{odd-row color}{even-row color}
```

where

row is the number of the first row to be colored,

odd-row color is the color to use with odd rows (if this parameter is blank it means no color),

even-row color is the color to use with even rows (if this parameter is blank it means no color).

Table 25: Table with colored odd rows obtained with the `xcolor` package.

D (in)	P_u (lbs)	u_u (in)	β	G_f (psi · in)
5	269.8	0.000674	1.79	0.04089
10	421.0	0.001035	3.59	0.04089
20	640.2	0.001565	7.18	0.04089

Table 26: Table with a colored cell obtained with the `\cellcolor` command.

Force	Force is a vector quantity defined as the rate of change of the momentum of the body that would be induced by that force acting alone.
Moment of a force	Moment of a force with respect to an origin is defined as the cross product of the position vector (with respect to the same origin) and the force.

For example table 25 is exactly the same as table 7 but, instead of horizontal lines (`\hline`), it uses gray background for odd rows. The code that is used to create this table is

```
\rowcolors{2}{gray!35}{}
\begin{tabular}{ccccc}
...
\end{tabular}
```

4.3 How to color single cells

The command `\cellcolor` can be used to add a background color to single cells: it works as `\columncolor` and `\rowcolor`, and prevails on both these. `\cellcolor` can be placed everywhere in the cells to which it has to be applied.

For example table 26 is obtained just adding the following command to the code of table 21,

```
\cellcolor[gray]{0.8}
```


5 Automatic generation of tables

Often the tables that appear in \LaTeX documents contain data generated by other programs, for example statistical or mathematical software. \LaTeX tables can be easily and automatically generated by other programs since their syntax is very simple. Typically the cells of a row are just separated by `&` and each row is ended by `\` (see par. 1.2.1).

5.1 Stata

An interesting application of this concept has been presented for the statistical program Stata.⁷ Gini [9] explains how to use Stata to automatically generate \LaTeX tables with colored rows (see par. 4) and tables in which some cells are highlighted with boldface (for example those that contains the maximum or the minimum of a series of data).

5.2 Spreadsheets

Many freeware softwares, as Excel2LaTeX⁸ and Spreadsheet2LaTeX⁹, that automatically generates \LaTeX code ready to be compiled starting from a formatted¹⁰ table in a spreadsheet. In general it is quite easy to program macros to make \LaTeX code starting from a generic spreadsheet.

6 Specialities

6.1 Line styles

As said in par. 4, the `colortbl` package can be used to modify the color of horizontal and vertical lines; anyway this should be avoided in scientific documents and so the details are left to the reader.

7. Stata is a registered trade mark of StataCorp LP.

8. Excel2LaTeX is available at <ftp://ftp.tex.ac.uk/tex-archive/support/excel2latex/xl2latex.zip>.

9. Spreadsheet2LaTeX is available at <http://pegasus.rutgers.edu/~elflord/unix/software/spreadsheet2latex/>.

10. For example with colors, horizontal and vertical lines.

The `arydshln` package allows to use dashed lines. The commands `\hdashline` and `\cdashline` corresponds respectively to `\hline` and `\cline`.

6.2 Notes inside a table

The standard L^AT_EX command for notes (`\footnote`) does not work inside tables because the `tabular` environment does not allow the command to place text at the bottom of the page. Many solutions and packages can be used to overcome this problem; the best ones are reported below.

In general, two cases can arise:

- a) the note must follow the numeration of the other notes in the text;
- b) the numeration of the notes inside tables must be different (for example with letters).

The package `footnote` provides a solution for case (a). This package defines the `savenotes` environment that allows to insert notes even inside a `tabular` environment and place them at the bottom of the page. If we want to allow the presence of notes in all the tables of the document, we just need to add to the preamble the command

```
\makesavenoteenv{tabular}
```

For example table 27 is obtained with the following code

```
\begin{savenotes}
\begin{table}[..] \caption{...} \label{...} \centering
\begin{tabular}{...} \toprule
...
Moment of a force & Moment of a force with respect to an origin
is defined as the cross product of the position vector\footnote{With
respect to the same origin.} and the force.\bottomrule
\end{tabular}
\end{table}
\end{savenotes}
```

The `ctable` package, on the other hand, offers a solution to case (b). The `ctable` environment, provided by the homonymic package, offers the command `\tmark` which places the symbol of the note (in this case the default numbering style is with lowercase letters) and the command `\tnote{...}` that contains the text of

11. With respect to the same origin.

Table 27: Table with a note that follows the numbering of the other notes in the text and is placed at the bottom of the page (footnote package).

Force	Force is a vector quantity defined as the rate of change of the momentum of the body that would be induced by that force acting alone.
Moment of a force	Moment of a force with respect to an origin is defined as the cross product of the position vector ¹¹ and the force.

Table 28: Table with its own numbering for the notes and with notes placed right after the table (ctable package).

Force	Force is a vector quantity defined as the rate of change of the momentum of the body that would be induced by that force acting alone.
Moment of a force	Moment of a force with respect to an origin is defined as the cross product of the position vector ^a and the force.

^a With respect to the same origin.

the note. The notes are placed right below the table and not at the bottom of the page. For example table 28 is obtained with the following code

```
\begin{ctable}%
[caption = ...,label = ...]{...}%
{\tnote{With respect to the same origin.}}%
{...\\midrule
Moment of a force & Moment of a force with respect to
an origin is defined as the cross product of the position
vector\tmark\ and the force.\\bottomrule}
\end{ctable}
```

Table 29: Table obtained with the slashbox package.

Function \ Argument	0	$\pi/2$
sin	0	1
cos	1	0

6.3 Slashed boxes

Sometimes the first cell on the top left of the table contains two arguments: the first one describes the content of the first column and the second one that of the first row. This solution, not recommended for scientific documents, can be obtained with the command `\backslashslashbox` provided by the slashbox package. For example the code

```
\begin{tabular}{|l|>{\$}c<{\$}|>{\$}c<{\$}|}\hline
\backslashslashbox{Function}{Argument} &0&\pi/2\\\hline
$\sin$ & 0 & 1\\\hline
$\cos$ & 1 & 0\\\hline
\end{tabular}
```

produces table 29.

The slashbox package draws a figure with two labels on the sides of an oblique line and then places this figure inside the cell of the table. The package uses L^AT_EX picture mode that has many restrictions and does not produce figures of high quality. The pict2e package can be used to improve the quality.

Acknowledgments

The author would like to thank Valeria Angeli for her assistance during the translation of this article from the italian version [19]. The author also would like to thank editorial staff of Ars T_EXnica, on which the original paper was published in Italian, that authorized the publication of this article on The PracT_EX Journal.

References

- [1] C. Beccari. *L^AT_EX, Guida a un sistema di editoria elettronica*. Hoepli, 1991.

- [2] D. P. Carlisle. *Packages in the 'graphics' bundle*, 1999. <ftp://tug.ctan.org/pub/tex-archive/macros/latex/required/graphics/grfguide.pdf>.
- [3] L. Caucci and M. Spadaccini. *Gestione di Figure e Tabelle con L^AT_EX*, 2005. http://www.guit.sssup.it/downloads/fig_tut.pdf.
- [4] P. W. Daly. *Graphics and Colour with L^AT_EX*, 1998. <http://tex.loria.fr/graph-pack/grf/grf.pdf>.
- [5] A. Diller. *L^AT_EX Line by Line: Tips and Techniques for Document Processing*. John Wiley & Sons, 1999.
- [6] R. Fairbairns. *The UK T_EX FAQ*, 2006. <http://www.ctan.org/tex-archive/help/uk-tex-faq/letterfaq.tex>.
- [7] S. Fear. *Publication quality tables in L^AT_EX*, 2005. <http://www.ctan.org/tex-archive/macros/latex/contrib/booktabs/booktabs.pdf>.
- [8] P. Flynn. *A beginner's introduction to typesetting with L^AT_EX*, 2005. <http://www.tug.org/tex-archive/info/beginlatex/>.
- [9] R. Gini. *Generazione automatica di tabelle con L^AT_EX e Stata*, 2004. http://www.guit.sssup.it/downloads/dispensa_gini.pdf.
- [10] M. Goossens, F. Mittelbach, and A. Samarin. *The L^AT_EX Companion*. Addison-Wesley, 1995.
- [11] G. Grätzer. *First Steps in L^AT_EX*. Springer Verlag, 1999.
- [12] G. Grätzer. *Math into L^AT_EX*. Birkhauser, 2000.
- [13] E. Gregorio. *L^AT_EX, Breve guida ai pacchetti di uso più comune*, December 2005. <http://profs.sci.univr.it/~gregorio/breveguida.pdf>.
- [14] J. Hahn. *L^AT_EX for Everyone: A Reference Guide and Tutorial for Typesetting Documents Using a Computer*. Prentice Hall, 1993.
- [15] D. J. Higham and D. F. Griffiths. *Learning L^AT_EX*. Society for Industrial and Applied Mathematics, 1997.
- [16] D. Knuth. *The T_EXbook*. Addison-Wesley, 1992.

- [17] H. Kopka and P. W. Daly. *A Guide to L^AT_EX – Document Preparation for Beginners and Advanced Users*. Addison-Wesley, 1995.
- [18] L. Lamport. *L^AT_EX: A Document Preparation System, User's Guide and Reference Manual*. Addison-Wesley, 1994.
- [19] L. F. Mori. Tabelle su L^AT_EX_{2 ϵ} —pacchetti e metodi da utilizzare. *Ars T_EXnica*, (2):31–47, October 2006.
- [20] L. F. Mori. Scrivere la tesi di laurea con L^AT_EX_{2 ϵ} . In *G_UITmeeting 2005*, Pisa, Italy, October 2005. G_UIT (Italian T_EX Users Group).
- [21] T. Oetiker, H. Partl, I. Hyna, and E. Schlegl. *Una (mica tanto) breve introduzione a L^AT_EX_{2 ϵ}* , 2000. <http://www.ctan.org/tex-archive/info/lshort/italian/itlshort.pdf>.
- [22] M. Trettin. *Elenco dei "peccati" degli utenti di L^AT_EX_{2 ϵ}* , 2005. <http://www.tug.org/tex-archive/info/l2tabu/italian/l2tabuit.pdf>.
- [23] H. Voß. *Rotating Text, Tabulars and Images*, 2003. <http://perce.de/LaTeX/rotating.pdf>.