Quick start for LaTeXing with the IEEEtran class for IEEE Computer Society Conferences

***Abstract*—Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.**

I. Introduction

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.[Todo 1: Refine me]

The remainder of the paper starts with a presentation of related work (Section II). It is followed by a presentation of hints on LaTeX (Section IV). The paper closes with a conclusion and an outlook on future work (Section V).

II. Related Work

Winery [1] is a graphical modeling [Todo 2: modeling with one “l”, because of American English] tool. The whole idea of TOSCA is explained by Binz et al. [2]. TOSCA is used in industry [Source 1: Use of TOSCA in industry].

In an anonymous submission (see `paper-anon.tex`), cite your own work in the third person, as any other work: “Binz et al. [2] showed”, not “in our previous work [2]”. Everything else that identifies the authors goes into `\anon`: this work was carried out at an anonymous university.

Tools for the literature search are, among others, ScienceOS, dblp, Google Scholar, IEEE Xplore, and the ACM Digital Library; JabRef manages the found papers, fetches the bibliographic data by DOI, and keeps the PDFs. If the search is systematic (as in a literature review), the keywords, the time span, and the inclusion criteria belong into the thesis so that a reader can repeat the search. Writing hints for student theses are given in the open-access book *Student Thesis Projects*[1]. More hints are given in *Writing for Computer Science*[2].

III. Writing in Markdown

This section uses plain Markdown, not LaTeX. Its source is the file `content-markdown.md`, which the main file includes with the `markdownInput` command of the `markdown` package[3]. This needs LuaLaTeX. Text can be *emphasized* or **bold**, and a citation looks like this [3]. Section IV-O shows more Markdown.

IV. LaTeX Hints

This section contains hints on writing LaTeX. It focuses on minimal examples, which can be directly adapted to the content

[1] https://vdf.ch/product/student-thesis-projects-en.html
[2] https://doi.org/10.1007/978-1-4471-6639-9
[3] https://ctan.org/pkg/markdown

A. *Handling of paragraphs*

One sentence per line. This rule is important for the usage of version control systems. A new line is generated with a blank line. As you would do in Word: New paragraphs are generated by pressing enter. In LaTeX, this does not lead to a new paragraph as LaTeX joins subsequent lines. To start a new paragraph, press enter twice. This leads to an empty line. In word, there is the functionality to press shift and enter. This leads to a hard line break. The text starts at the beginning of a new line. In LaTeX, you can do that by using two backslashes (\\).
Running text seldom needs a hard line break.

Please do *not* use two backslashes for new paragraphs. For instance, this sentence belongs to the same paragraph, whereas the last one started a new one. Andrew Stacey gives a long motivation for that in the section "Line Wrapping" of his *Document Revision System*[4].

Corresponding LaTeX code of `paper.tex`

```
852 One sentence per line.
853 This rule is important for the usage of
    ↪ version control systems.
854 A new line is generated with a blank
    ↪ line.
855 As you would do in Word:
856 New paragraphs are generated by pressing
    ↪ enter.
857 In LaTeX, this does not lead to a new
    ↪ paragraph as LaTeX joins subsequent
    ↪ lines.
858 To start a new paragraph, press enter
    ↪ twice.
859 This leads to an empty line.
860 In word, there is the functionality to
    ↪ press shift and enter.
861 This leads to a hard line break.
862 The text starts at the beginning of a new
    ↪ line.
863 In LaTeX, you can do that by using two
    ↪ backslashes
    ↪ (\textbackslash\textbackslash).
864 \\
865 Running text seldom needs a hard line
    ↪ break.
866
867 Please do \textit{not} use two
    ↪ backslashes for new paragraphs.
868 For instance, this sentence belongs to
    ↪ the same paragraph, whereas the last
    ↪ one started a new one.
869 Andrew Stacey gives a long motivation for
    ↪ that in the section \enquote{Line
    ↪ Wrapping} of his \href{https://
    ↪ loopspace.mathforge.org/
    ↪ HowDidIDoThat/TeX/VCS/#section.3}{
    ↪ \emph{Document Revision
    ↪ System}}\footnote{\url{https://
    ↪ loopspace.mathforge.org/
    ↪ HowDidIDoThat/TeX/VCS/}}.
```

Once the text is final, single paragraphs can be polished. If the last line of a paragraph holds only a word or two, or is nearly full, let TeX set the paragraph one line shorter or longer: write `\looseness=-1` (or `=1`) as the first thing in that paragraph. TeX resets `\looseness` at the end of every paragraph, so only that one paragraph is affected. With LuaLaTeX, a lone short word as the last line is avoided automatically (package impnattypo).

[4] https://loopspace.mathforge.org/HowDidIDoThat/TeX/VCS/

B. *Notes separated from the text*

The package mindflow enables writing down notes and annotations in a way so that they are separated from the main text.

This is a small note.

Corresponding LaTeX code of `paper.tex`

```
882 \begin{mindflow}
883 This is a small note.
884 \end{mindflow}
```

C. *Handling TODOs*

Marked text.

Corresponding LaTeX code of `paper.tex`

```
890 \textmarker{Marked text.}
```

With `\textmarker`, only the text color is changed, since this also works well for individual words.
Marked text. [Todo 3: A comment on it.] 3

Corresponding LaTeX code of `paper.tex`

```
896 \textcomment{Marked text.}{A comment on
    ↪ it.}
```

Manual marking for text that has been changed since the last version.

Corresponding LaTeX code of `paper.tex`

```
900 \modified{Manual marking for text that has
    ↪ been changed since the last version.}
```

This is some text. FL1: text adjusted → Changed text.

Corresponding LaTeX code of `paper.tex`

```
904 This is some text.
905 \change{FL1: text adjusted}{Changed
    ↪ text}.
```

Here is a comment[Comment 1: Comment]. 1

Corresponding LaTeX code of `paper.tex`

```
909 Here is a comment\sidecomment{Comment}.
```

4

[Todo 4: A lot of text still needs to be produced here]

Corresponding LaTeX code of paper.tex

```
\todo{A lot of text still needs to be produced here}
```

\TODO, \FIXME, and \CITE work with every package for notes the template offers: \TODO marks an open task, \FIXME an important one, and \CITE a claim that still needs a citation.

[Todo 5: Add the evaluation]

[Fixme 1: The numbers are outdated]

Most scientific texts in computer science are written with LaTeX [Source 2: Survey on the tools used for writing].

Corresponding LaTeX code of paper.tex

```
\TODO{Add the evaluation}

\FIXME{The numbers are outdated}

Most scientific texts in computer science are written with \LaTeX{}~\CITE{Survey on the tools used for writing}.
```

With zebra, \todo takes the responsible person as optional argument; further note types are \note, \comment, \fixed, \placeholder, \fixme, and \source (defined by the template, used by \FIXME and \CITE). All notes are listed at the end of the document.

[Todo 6@FA: Add a citation]

[Note 1: Shorten this paragraph?]

[Comment 2@FA: Is this claim backed by the evaluation?]

[Fixed 1: Typo corrected]

[Placeholder 1: Table with the results]

Corresponding LaTeX code of paper.tex

```
\todo[FA]{Add a citation}

\note{Shorten this paragraph?}

\comment[FA]{Is this claim backed by the evaluation?}

\fixed{Typo corrected}

\placeholder{Table with the results}
```

Co-authors can mark their edits and comments with the phfcc package; the initials of the author appear once per page in the margin. Define one command per author in the preamble (\phfMakeCommentingCommand); \phfDisableCommentingCommands hides everything for the final version.

Quantum mechanics developed within one decade thanks to the works of Schrödinger, Heisenberg, and Dirac, who drew inspiration from earlier ideas by Planck and Einstein [Einstein must not be missing here.] ~~This sentence is to be removed.~~ **!!! This is important. !!!**

Corresponding LaTeX code of paper.tex

```
Quantum mechanics developed within one decade thanks to the works of Schrödinger, Heisenberg, and Dirac, who drew inspiration from earlier ideas by Planck \FA{and Einstein} \FA[Einstein must not be missing here.] \FA*{This sentence is to be removed.} \FA!{This is important.}
```

D. Hyphenation

LaTeX automatically hyphenates words. When using microtype, there should be fewer hyphenations than in other settings. You might still need to tweak the hyphenation. Here are some hints:

In case you write “application-specific”, then the word will only be hyphenated at the dash. You can also insert \allowbreak{} into the word, as in application-specific, but this is much more effort.

You can now write words containing hyphens which are hyphenated at other places in the word. For that, write "= instead of −, as in application"=specific. This is enabled by an additional configuration of the babel package.

Corresponding LaTeX code of paper.tex

```
In case you write \enquote{application-specific}, then the word will only be hyphenated at the dash.
You can also insert \verb1\allowbreak{}1 into the word, as in applica\allowbreak{}tion-specific, but this is much more effort.

You can now write words containing hyphens which are hyphenated at other places in the word.
For that, write \verb1"=1 instead of \verb1-1, as in application"=specific. % chktex 18
This is enabled by an additional configuration of the babel package.
```

E. Typesetting Units

Numbers can be written plain text (such as 100), by using the siunitx package as follows: 100 $\frac{\mathrm{km}}{\mathrm{h}}$, or by using plain LaTeX (and math mode): $100\frac{km}{h}$.

Corresponding LaTeX code of paper.tex

```
Numbers can be written plain text (such as 100), by using the \href{https://ctan.org/pkg/siunitx}{siunitx} package as follows:
\SI{100}{\km\per\hour},
or by using plain \LaTeX{} (and math mode):
$100 \frac{\mathit{km}}{h}$.
```

5 % of 10 kg

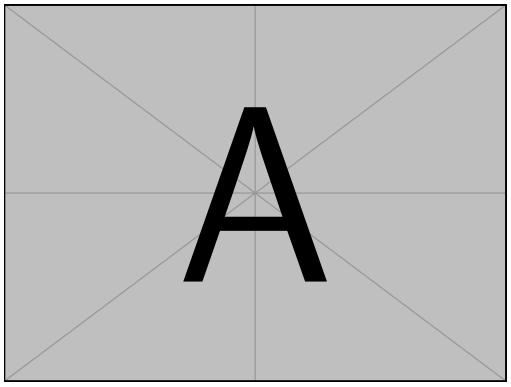


Fig. 1. Example figure for zcref demo

Table I
EXAMPLE TABLE FOR ZCREF DEMO

Heading1	Heading2
One	Two
Three	Four

Corresponding LaTeX code of `paper.tex`

```
977 \SI{5}{\percent} of \SI{10}{kg}
```

Numbers are automatically grouped: 123 456.

Corresponding LaTeX code of `paper.tex`

```
981 Numbers are automatically grouped:
    ↪ \num{123456}.
```

F. Surrounding Text by Quotes

Please use the “enquote command” to quote something. Quoting with “quote” or “quote” also works.

Corresponding LaTeX code of `paper.tex`

```
987 Please use the \enquote{enquote command}
    ↪ to quote something.
988 Quoting with "`quote"' or ``quote'' also
    ↪ works. % chktex 18
989
```

G. Zref-clever examples

Demonstration of zref-clever: Zcref (short for zcref[S]) at the beginning of a sentence, zcref in all other cases.

Fig. 1 shows a simple fact, although Fig. 1 could also show something else.

Table I shows a simple fact, although Table I could also show something else.

Section IV-G shows a simple fact, although Section IV-G could also show something else.

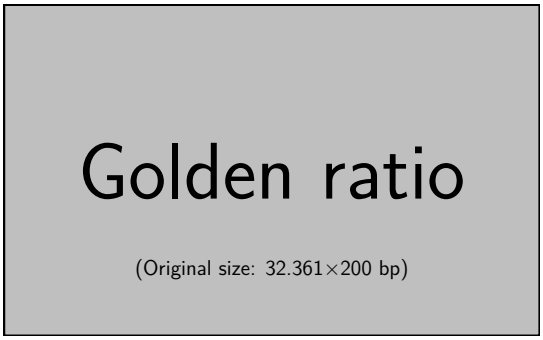


Fig. 2. Simple Figure. Based on Scharrer [3].

Corresponding LaTeX code of `paper.tex`

```
1019 \Zcref{fig:ex:cref} shows a simple fact,
     ↪ although \zcref{fig:ex:cref} could
     ↪ also show something else.
1020
1021 \Zcref{tab:ex:cref} shows a simple fact,
     ↪ although \zcref{tab:ex:cref} could
     ↪ also show something else.
1022
1023 \Zcref{sec:ex:cref} shows a simple fact,
     ↪ although \zcref{sec:ex:cref} could
     ↪ also show something else.
```

H. Figures

Fig. 2 shows something interesting.

Corresponding LaTeX code of `paper.tex`

```
1029 \Zcref{fig:label} shows something
     ↪ interesting.
1030
1031 \begin{figure}
1032   \centering
1033   \includegraphics[width=.8\linewidth]{
       ↪ example-image-golden}
1034   \caption[Simple Figure]{
1035     Simple Figure.
1036     Based on \citet{mwe}.
1037   }
1038   \label{fig:label}
1039 \end{figure}
```

Photos in JPEG or PNG format are included the same way, as shown in Fig. 3.

Corresponding LaTeX code of `paper.tex`

```
1045 \begin{figure}
1046   \centering
1047   \includegraphics[width=.5\linewidth]{
       ↪ figures/ulm-minster.jpg}
1048   \caption{Ulm Minster. Photo: wkrauth,
       ↪ CC0, via Wikimedia Commons.}
1049   \label{fig:jpg}
1050 \end{figure}
```

A figure spans multiple columns when using `\begin{figure*}`. See Fig. 4 as an example.



Fig. 3. Ulm Minster. Photo: wkrauth, CC0, via Wikimedia Commons.

Corresponding LaTeX code of `paper.tex`

```
\begin{figure*}
  \centering
  % note that \textwidth is used instead of \linewidth
  % This ensures that the graphics width is 60% of the "page" (text block), and not just 60% of the current text column
  % See https://tex.stackexchange.com/a/17085/9075 for details
  \includegraphics[width=.6\textwidth]{example-image-16x9}
  \caption{Figure in 16:9 format}
  \label{fig:16x9}
\end{figure*}
```

I. Sub Figures

An example of two sub figures is shown in Fig. 5.

Corresponding LaTeX code of `paper.tex`

```
\begin{figure*}[!t]
  \centering
  \subfloat[Case I]{\includegraphics[width=.4\linewidth]{example-image-a}%
    \label{fig:first_case}}
  \hfil
  \subfloat[Case II]{\includegraphics[width=.4\linewidth]{example-image-b}%
    \label{fig:second_case}}
  \caption{Example figure with two sub figures.}
  \label{fig:two_sub_figures}
\end{figure*}
```

Note that often IEEE papers with subfigures do not employ subfigure captions (using the optional argument to `\subfloat[]`), but instead will reference/describe all of them (a), (b), etc., within the main caption. Be aware that for the subfig package to generate the (a), (b), etc., subfigure labels, the optional argument to `\subfloat` must be present. If a subcaption is not desired, leave its contents blank, e.g., `\subfloat[]`. An example is shown in Fig. 6.

Corresponding LaTeX code of `paper.tex`

```
\begin{figure*}[!t]
  \centering
  \subfloat[]{\includegraphics[width=.4\linewidth]{example-image-a}%
    \label{fig:first_case_ieee}}
  \hfil
  \subfloat[]{\includegraphics[width=.4\linewidth]{example-image-b}%
    \label{fig:second_case_ieee}}
  \caption{Example figure with two sub figures in IEEE style: (a) the first case, (b) the second case.}
  \label{fig:two_sub_figures_ieee}
\end{figure*}
```

J. Figures with tikz

The tikz package creates graphics programmatically. This package generates grids and other regular structures.

Corresponding LaTeX code of `paper.tex`

```
\begin{figure}[ht]
  \centering
  \begin{tikzpicture}
    \draw(0,0) rectangle (4,4);
    \foreach \x in {0.5,1,1.5,2,2.5,3,3.5}
    \foreach \y in {0.5,1,1.5,2,2.5,3,3.5}
    \draw(\x,\y) circle (1pt);
  \end{tikzpicture}
  \caption{A regular grid generated easily with two for loops.}\label{fig:tikz_example}
\end{figure}
```

K. Plots with pgfplots

The pgfplots package plots functions directly in LaTeX, similar to MATLAB or gnuplot. Some visual examples are available in the package documentation[5].

Corresponding LaTeX code of `paper.tex`

```
\begin{figure}[ht]
  \centering
  \begin{tikzpicture}
    \begin{axis}[xlabel=$x$, ylabel=$\sin(x)$]
      \addplot {sin(deg(x))};  % Plot the sine function
    \end{axis}
  \end{tikzpicture}
  \caption{Plot of $\sin(x)$ directly inside the figure environment with pgfplots.}
  \label{fig:pgfplots_example}
\end{figure}
```

Coordinates can also be given explicitly instead of as a function:

[5] http://texdoc.net/pkg/pgfplots

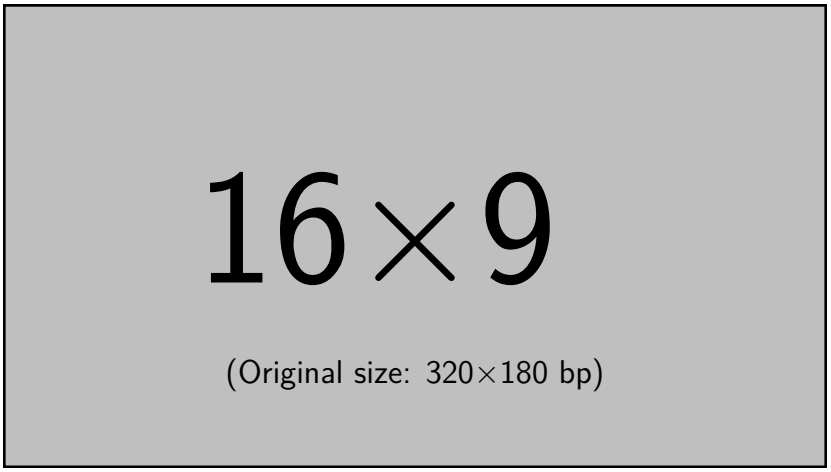
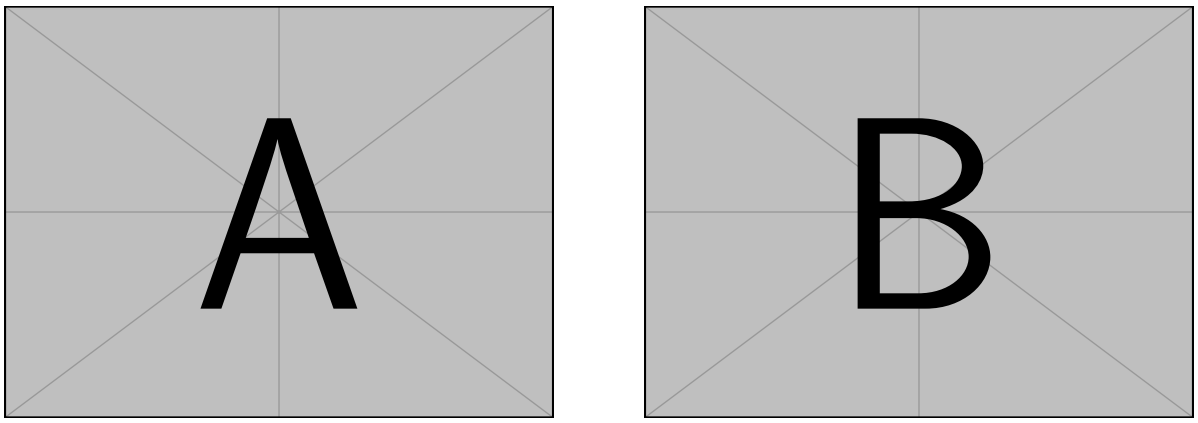
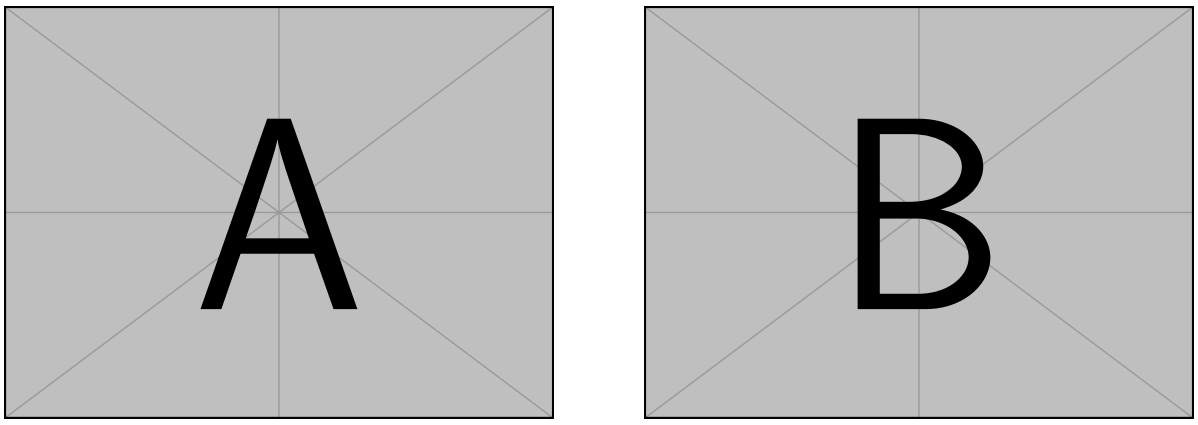


Fig. 4. Figure in 16:9 format



(a) Case I (b) Case II

Fig. 5. Example figure with two sub figures.



(a) (b)

Fig. 6. Example figure with two sub figures in IEEE style: (a) the first case, (b) the second case.

Corresponding LaTeX code of `paper.tex`

```
1146 \begin{figure}[ht]
1147     \centering
1148     \begin{tikzpicture}
```

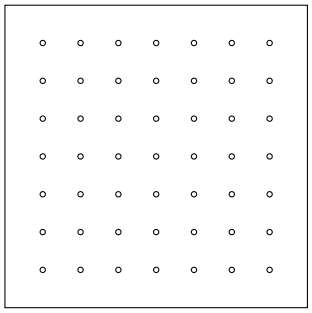


Fig. 7. A regular grid generated easily with two for loops.

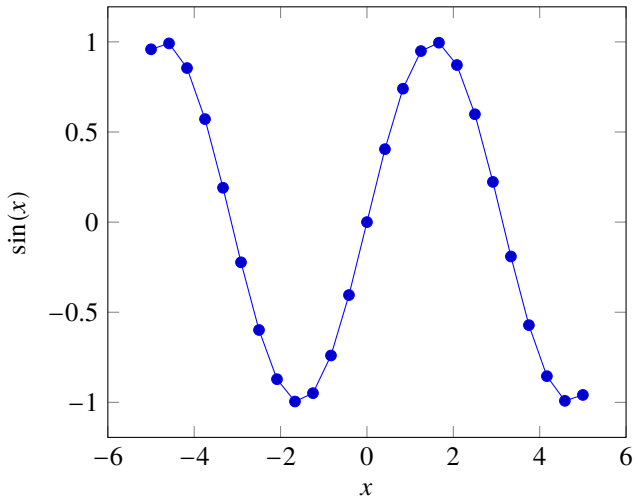


Fig. 8. Plot of $\sin(x)$ directly inside the figure environment with pgfplots.

```
    \begin{axis}[xlabel=$x$, ylabel=$y$]
      \addplot coordinates {(0,0) (1,1)
        ↪ (2,4) (3,9)};  % Plot explicit
        ↪ coordinates
    \end{axis}
  \end{tikzpicture}
  \caption{Explicit coordinates plotted
    ↪ with pgfplots.}
  \label{fig:pgfplots_coords}
\end{figure}
```

L. Tables

Corresponding LaTeX code of `paper.tex`

```
\begin{table}
  \caption{Simple Table}
  \label{tab:simple}
  \centering
```

Table II
SIMPLE TABLE

Heading1	Heading2
One	Two
Three	Four

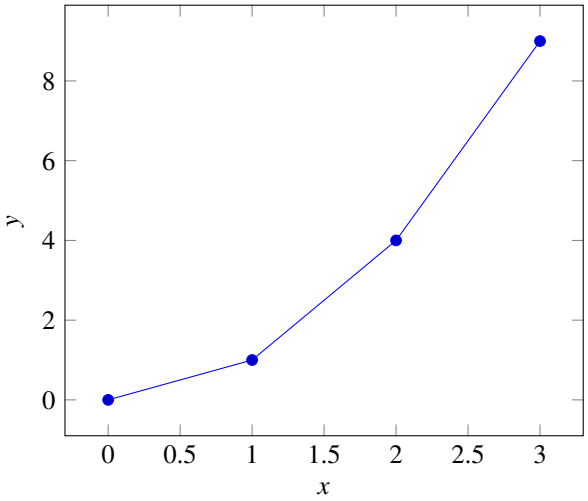


Fig. 9. Explicit coordinates plotted with pgfplots.

Table III
TABLE WITH COLORED CELLS

Test	Result
A	pass
B	fail

```
  \begin{tabular}{ll}
    \toprule
    Heading1 & Heading2 \\
    \midrule
    One      & Two      \\
    Three    & Four     \\
    \bottomrule
  \end{tabular}
\end{table}
```

A table caption goes above the table (`\caption` before the `tabular`), a figure caption below the figure.

Rows and cells can be colored with `\rowcolor` and `\cellcolor` from the colortbl package, which xcolor loads with its option `table`.

Corresponding LaTeX code of `paper.tex`

```
\begin{table}
  \caption{Table with colored cells}
  \label{tab:colored}
  \centering
  \begin{tabular}{ll}
    \toprule
    \rowcolor{gray!20}
    Test & Result                    \\
    \midrule
    A    & \cellcolor{green!20}pass \\
    B    & \cellcolor{red!20}fail   \\
    \bottomrule
  \end{tabular}
\end{table}
```

A `\footnote` inside a floating table is lost. Use `\tablefootnote` from the tablefootnote package instead: it

Table IV
TABLE WITH A FOOTNOTE

Heading1	Heading2[6]
One	Two

Table V
WIDE TABLE SCALED TO THE LINE WIDTH

Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
1.11 Rain	2.22 Snow	3.33 Wind	4.44 Sun	5.55 Fog	6.66 Hail	7.77 Heat	8.88 Storm	9.99 Mist	10.10 Frost	11.11 Sleet	12.12 Calm

is set as a normal footnote at the bottom of the page. If the notes should appear directly below the table instead (common in journals), use the threeparttable package.

Corresponding LaTeX code of `paper.tex`

```
\begin{table}
  \caption{Table with a footnote}
  \label{tab:footnote}
  \centering
  \begin{tabular}{ll}
    \toprule
    Heading1 & Heading2\tablefootnote{A footnote inside a table.} \\
    \midrule
    One      & Two \\
    \bottomrule
  \end{tabular}
\end{table}
```

A table wider than the text runs into the margin. `\begin{adjustbox}{max width=\linewidth}` from the adjustbox package scales it down to the line width (the column width in a two-column layout), and only if it is too wide. Scaling shrinks the font, so for a table with long text in its cells, prefer wrapping the text with the column type `X` of the tabularx package, and for a table wider than the page, a landscape page (see `pdflscape`).

Corresponding LaTeX code of `paper.tex`

```
\begin{table}
  \caption{Wide table scaled to the line width}
  \label{tab:wide}
  \centering
  \begin{adjustbox}{max width=\linewidth}
    \begin{tabular}{llllllllllll}
      \toprule
      Jan  & Feb  & Mar  & Apr  & May  & Jun  & Jul  & Aug  & Sep  & Oct  & Nov  & Dec  \\
      \midrule
      1.11 & 2.22 & 3.33 & 4.44 & 5.55 & 6.66 & 7.77 & 8.88 & 9.99 & 10.10 & 11.11 & 12.12 \\
      Rain & Snow & Wind & Sun  & Fog  & Hail & Heat & Storm & Mist & Frost & Sleet & Calm \\
      \bottomrule
    \end{tabular}
  \end{adjustbox}
\end{table}
```

[6]A footnote inside a table.

Table VI
TABLE WITH DIAGONAL LINE

Row \ Column	A	B
1	x	y

Corresponding LaTeX code of `paper.tex`

```
% Adapted from https://tex.stackexchange.com/a/468994/9075
\begin{table}
  \caption{Table with diagonal line}
  \label{tab:diag}
  \centering
  \begin{tabular}{|l|c|c|} % chktex 44
    \hline % chktex 44
    \diagbox{Row}{Column} & A & B \\
    \hline % chktex 44
    1                     & x & y \\
    \hline % chktex 44
  \end{tabular}
\end{table}
```

M. Source Code

The minted package highlights listings properly. It uses the pygments library, which in turn requires Python.

Listing 1 shows source code written in XML. Line 2 contains a comment.

```
1 <listing name="example">
2   <!-- comment -->
3   <content>not interesting</content>
4 </listing>
```

List. 1. Example XML listing using minted

Corresponding LaTeX code of `paper.tex`

```
% After a period, LaTeX sets the wider sentence space by default. The exception: a period after a capital letter
% is taken for an abbreviation or initial ("T. Scientist", "NASA") and gets the narrow inter-word space.
% "XML." ends a sentence, so \@ before the period restores the sentence space.
% (The other way round, "e.g.\ details" forces the narrow space after a lowercase abbreviation; ~ is a
% non-breaking inter-word space, as in "Prof.~Dr.~Eike Kraft".)
```

```
% An abbreviation set with \gls{xml} needs no \@: glossaries ends it with a non-letter, so the period after it gets the sentence space. The plain "XML" here is for documents without the glossaries package.
\Zcref{lst:XML} shows source code written in XML\@.
\Zcref{line:comment} contains a comment.

\begin{listing}[htbp]
  \begin{minted}[linenos=true,escapeinside=||]{xml}
<listing name="example">
  <!-- comment --> |\label{line:comment}|
  <content>not interesting</content>
</listing>
\end{minted}
  \caption{Example XML listing using minted}
  \label{lst:XML}
\end{listing}
```

One can also typeset JSON as shown in Listing 2.

```
{
  key: "value"
}
```

List. 2. Example JSON listing using minted

Corresponding LaTeX code of `paper.tex`

```
\begin{listing}[htbp]
  \begin{minted}[linenos=true,escapeinside=||]{json}
{
  key: "value"
}
\end{minted}
  \caption{Example JSON listing using minted}
  \label{lst:flJSON}
\end{listing}
```

Java is also possible as shown in Listing 3.

```
public class Hello {
    public static void main (String[] args) {
        System.out.println("Hello World!");
    }
}
```

List. 3. Java code rendered using minted

Corresponding LaTeX code of `paper.tex`

```
\begin{listing}[htbp]
  \begin{minted}[linenos=true,escapeinside=||]{java}
public class Hello {
    public static void main (String[] args) {
        System.out.println("Hello World!");
    }
}
\end{minted}
  \caption{Java code rendered using minted}
  \label{lst:flJava}
\end{listing}
```

N. *Itemization*

One can list items as follows:

- Item One
- Item Two

Corresponding LaTeX code of `paper.tex`

```
\begin{itemize}
  \item Item One
  \item Item Two
\end{itemize}
```

With the package paralist, one can create itemizations with lesser spacing:

- Item One
- Item Two

Corresponding LaTeX code of `paper.tex`

```
\begin{compactitem}
  \item Item One
  \item Item Two
\end{compactitem}
```

One can enumerate items as follows:

1) Item One
2) Item Two

Corresponding LaTeX code of `paper.tex`

```
\begin{enumerate}
  \item Item One
  \item Item Two
\end{enumerate}
```

With the package paralist, one can create enumerations with lesser spacing:

1) Item One
2) Item Two

Corresponding LaTeX code of `paper.tex`

```
\begin{compactenum}
  \item Item One
  \item Item Two
\end{compactenum}
```

With paralist, one can even have all items typeset after each other and have them clean in the TeX document:

1) The first item, 2) the second item, and 3) the third item appear in one line.

Table VII
A TABLE WRITTEN IN MARKDOWN

Right	Left	Centered
12	12	12
123	123	123

Corresponding LaTeX code of `paper.tex`

```
\begin{inparaenum}
  \item The first item,
  \item the second item, and
  \item the third item appear in one
   ↪ line.
\end{inparaenum}
```

O. Writing in Markdown

If you are new to LaTeX, you can start by writing your text in Markdown using the markdown package. LaTeX then takes care of the typesetting. Markdown and LaTeX can be mixed: Table VII is written in Markdown and referenced from LaTeX. A whole section can be written in a separate Markdown file included by `\markdownInput`, as Section III shows. The markdown-latex-quickstart shows a complete document written in Markdown.

Text can be *emphasized* or **bold**, and `code` is set in a monospaced font. A citation looks like this [3], and here is a footnote.[7] Referenced footnotes work as well[8].

- first point
- second point
  - sub-point

1. firstly
2. secondly

Table VII shows a table with caption. Its identifier `{#tbl:markdown}` is referenced by `<#tbl:markdown>`.

Corresponding LaTeX code of `paper.tex`

```
  \begin{markdown}
Text can be *emphasized* or **bold**, and
 ↪ `code` is set in a monospaced font.
A citation looks like this [@mwe], and
 ↪ here is a footnote.^[This is an
 ↪ inline footnote.]
Referenced footnotes work as
 ↪ well[^detail].

[^detail]: This is a footnote defined at
 ↪ a different place in the text.

- first point
- second point
  - sub-point

1) firstly
2) secondly

| Right | Left | Centered |
|------:|:-----|:--------:|
|    12 | 12   |    12    |
|   123 | 123  |   123    |

: A table written in Markdown
 ↪ {#tbl:markdown}

<#tbl:markdown> shows a table with
 ↪ caption.
Its identifier `{#tbl:markdown}` is
 ↪ referenced by `<#tbl:markdown>`.
\end{markdown}
```

[7]This is an inline footnote.
[8]This is a footnote defined at a different place in the text.

P. Other Features

The words “workflow” and “dwarflike” can be copied from the PDF and pasted to a text file.

Corresponding LaTeX code of `paper.tex`

```
The words \enquote{workflow} and
 ↪ \enquote{dwarflike} can be copied from
 ↪ the PDF and pasted to a text file.
```

The symbol for powerset is now correct: $\wp$ and not a Weierstrass p ($\wp$).
$\wp(1,2,3)$

Corresponding LaTeX code of `paper.tex`

```
The symbol for powerset is now correct:
 ↪ $\powerset$ and not a Weierstrass
 ↪ p ($\wp$).

$\powerset({1,2,3})$
```

Brackets work as designed: <test> One can also input backticks in verbatim text: `` `test` ``.

Corresponding LaTeX code of `paper.tex`

```
Brackets work as designed:
<test>
One can also input backticks in verbatim
 ↪ text: \verb|`test`|.
```

Acronyms can be typeset in running text with `\initialism`, which sets them as small caps that blend into the surrounding lowercase text. It is the lightweight alternative to the full acronym and glossary management (`\gls`, with an automatic list of abbreviations) provided by the thesis variants: `\initialism` only typesets the acronym and does not add it to any list. The macro and the ready-made acronyms (`\OMG`, `\BPEL`, `\BPMN`, `\UML`) are defined in `commands.tex`, where you can add your own.

The UML is maintained by the OMG; related standards include BPEL and BPMN. Any acronym can be set inline with REST or HTTP.

Corresponding LaTeX code of `paper.tex`

```
1423 The \UML is maintained by the \OMG;
     ↪ related standards include \BPEL and
     ↪ \BPMN.
1424 Any acronym can be set inline with
     ↪ \initialism{REST} or
     ↪ \initialism{HTTP}.
```

V. Conclusion and Outlook

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Acknowledgment

References

[1] O. Kopp *et al.*, "Winery – A Modeling Tool for TOSCA-based Cloud Applications," in *Proceedings of 11th International Conference on Service-Oriented Computing (ICSOC'13)*, LNCS, vol. 8274. Springer Berlin Heidelberg, 2013, pp. 700–704, doi: 10.1007/978-3-642-45005-1_64.

[2] T. Binz, G. Breiter, F. Leymann, and T. Spatzier, "Portable Cloud Services Using TOSCA," *IEEE Internet Computing*, vol. 16, no. 03, pp. 80–85, May 2012, doi: 10.1109/mic.2012.43.

[3] M. Scharrer. *The mwe Package*. (2017). [Online]. Available: http://texdoc.net/mwe

All links were last followed on September 25, 2027.

Zebra Notes

Type	Count
todo	6
fixed	1
comment	2
note	1
placeholder	1
fixme	1
source	2
Total	**14**

List of notes

Todo 1 . . . 1
Refine me

Todo 2 . . . 1
modeling with one "l", because of American English

Source 1 . . . 1
Use of TOSCA in industry

Todo 3 . . . 2
A comment on it.

Comment 1 . . . 2
Comment

Todo 4 . . . 2
A lot of text still needs to be produced here

Todo 5 . . . 3
Add the evaluation

Fixme 1 . . . 3
The numbers are outdated

Source 2 . . . 3
Survey on the tools used for writing

Todo 6 `@FA` . . . 3
Add a citation

Note 1 . . . 3
Shorten this paragraph?

Comment 2 `@FA` . . . 3
Is this claim backed by the evaluation?

Fixed 1 . . . 3
Typo corrected

Placeholder 1 . . . 3
Table with the results