Boxes and more boxes
Ivan R. Pagnossin

Abstract
Boxes are a key concept in L\TeX as well as an useful tool, though not a very well known one. I believe this is due to the success of L\TeX in providing a high level interface to\TeX, which allows the user to produce beautiful documents without getting acquainted with the concept of boxes. But it is useful to know the box concept because it enhances our understanding of \TeX, helps us to avoid and solve problems and can even propose solutions for unusual circumstances.

In this paper we reproduce, step by step, the reflection effect in the title of this article in order to gain some knowledge of the box concept.

1 Box basics
Among the fundamental ingredients of a L\TeX document, there are boxes: imaginary rectangles which enclose letters, lines, pages, paragraphs, figures, tables etc. Indeed, L\TeX does not typeset letters or characters, lines . . . but boxes!

You may have already used boxes to avoid breaking some word or maybe to frame an equation. But they are far more useful than this. A simple but instructive example is the reflection effect in the title of this paper, which requires one customized box. Let’s learn something about it.

Figure 1 shows the box associated with the letter g. It has three dimensions: height, width and depth. It is these dimensions that L\TeX actually cares about, not the content of the box. In other words, L\TeX does not “see” the letter g, only its box. This is the important fact to understand.

We can ask L\TeX to show us this box: just write \framebox{g}. Well, first we must set \fboxsep to zero, in order to remove the extra space around g: \setlength{\fboxsep}{0in}.

Much more important than just showing the box, \framebox builds a box with whatever you want inside it. For instance, \framebox{Boxes}. Now, Boxes is a single box, as unbreakable as the letter g itself. By the way, this is another property to keep in mind: boxes cannot be broken.

Another interesting property is this: the contents of a box need not lie inside it. You may have noticed that, given the contents as an argument, the \framebox command sets the dimensions of the box to those of the contents (in reality, to the “sub-boxes” that compose the contents). But you can define the dimensions explicitly as well. For example,

\framebox{.67in}{Boxes}

produces \framebox{Boxes}, a box \frac{3}{4} in wide. That is, a box which occupies more space than its contents. You can also define the alignment of the content with the box, to the left, center or right, as shown below:

- \framebox{.67in}{[l]Boxes}
- \framebox{.67in}{[c]Boxes}
- \framebox{.67in}{[r]Boxes}

In a similar fashion, we may create a box which occupies less space than its contents; a zero-width box, for instance:

\framebox{[0in]{Boxes}}

Try it yourself and see that Boxes overlaps the text at its right. This happens because, as we have seen, L\TeX uses the dimensions of the letters (boxes) to place them side by side in a row. However, the box we have just created occupies no (horizontal) space. It looks like it is not there, though its contents are (try also changing the alignment parameter).

2 Reflecting, scaling, coloring
Another command we will need to produce the reflection effect is \scalebox, defined in the package graphics. Its syntax is \scalebox{\textwidth}{f}{a box}. It changes the dimensions of the box and its contents according to the horizontal and vertical scaling factors \textwidth and \textwidth (respectively) and creates a new box that encloses this new content. Thus, as a first step toward our goal, to get Boxes we write

\scalebox{1}{-1}{Boxes}

This means we instruct L\TeX to multiply the width by \textwidth (hence, nothing changes on the horizontal) and the height (> 0 in our case) and the depth ( = 0) by \textwidth. As a result, we have a vertical reflection of Boxes. A hint: the commands defined by graphics receive boxes as arguments, not just figures (which are boxes too). Since the package was designed to handle figures, it is common to think
that its commands apply only to figures, but this is not true.

Let’s proceed. How can we place \textcolor{red!80}{Boxes} below \textcolor{red!80}{Boxes}? Answer: by constructing it in a way that occupies no horizontal space, that is, by putting it in a zero-width box (new code in black, previous code in gray):

\makebox[0in][l]{\scalebox{1}[{-1}]\{\textcolor{red!80}\{Boxes\}\{Boxes\}}

We have also swapped \framebox for \makebox. These two commands are equivalent, except that the former draws a frame around the box, while the latter does not.

Finally, we choose the reflection color (use the package xcolor):

\makebox[0in][l]{\scalebox{1}[-1]{\{\textcolor{red!80}{}\{Boxes\}\{Boxes\}}}

The sequence below illustrates the changing of the box width, from its natural width (defined by the content) down to zero. Notice that the non-inverted \textcolor{red!80}{Boxes} always starts immediately after the frame (the box), not after the content (\textcolor{red!80}{Boxes}).

3 Boxes upon boxes

There are boxes far more complex than these. As I mentioned, everything that is visible in a document produced by \LaTeX{} is in one or more boxes. There are even invisible elements in your document that are boxes too. An example is the paragraph indentation, which is nothing more than an empty box of width \parindent. In all cases, all of the concepts we have just seen remain valid.

Let’s see an example. The line below contains three boxes plus some filling lines (which are boxes too): the first is an entire paragraph, the second is a figure and the third a mathematical table (or matrix, if you prefer). All of them have height, width and depth, are unbreakable, and are placed side by side with their reference points aligned (fig. 1) over an imaginary line called the baseline, represented by the horizontal filling lines.

\begin{verbatim}
\setlength{\parindent}{1em}
\begin{minipage}[t]{0.16\textwidth}
\fbox{\parbox{0.16\textwidth}{Just a short example paragraph. A line or two or three. That is all.}}
\end{minipage}
\end{verbatim}

The paragraph box is aligned on the baseline of its first line. The figure, on the other hand, inserted with the \includegraphics command (package graphicx), has depth zero (but since we’ve enclosed it in an \fbox for the example, it is not quite zero here). Finally, the matrix has unequal height and depth (it’s centered on the math axis; the \LaTeX{} code that produces this line is at the end of this paper). The important thing here is to notice that the placement of all these boxes follow exactly the same rules as those followed by letters, since they are all in boxes.

So, whenever you insert a figure or table in your document, see them as “big \textcolor{red!80}{g} letters”. In other words, see the boxes!

4 Further reading

To learn more about this subject, you can study the environment minipage, the commands \vspace, \includegraphics and \setlength in section 4.7 of [1] and/or appendix A.2 of [2]. Chapter 11 of [3] is mandatory. Moreover, have a look at the commands defined by the package graphicx.

Appendix: Our unusual line of text

The code used to produce the unusual text line discussed above in this paper is shown below. You will need the packages graphicx and amsmath to run it.

\begin{verbatim}
\setlength{\fboxsep}{0pt}
\begin{minipage}{0.16\textwidth}
\fbox{$\cos \varphi \quad -\sin \varphi \quad 0$
  \parbox{0.16\textwidth}{\scalebox{1}[{-1}]\{\textcolor{red!80}\{Boxes\}\{Boxes\}}}$
  \parbox{0.16\textwidth}{\scalebox{1}[{-1}]\{\textcolor{red!80}\{Boxes\}\{Boxes\}}}
\end{minipage}
\end{verbatim}

\begin{verbatim}
\begin{matrix}
\cos \varphi & -\sin \varphi & 0 \\
\sin \varphi & \cos \varphi & 0 \\
0 & 0 & 1
\end{matrix}
\end{verbatim}

Acknowledgments

The author thanks Karl Berry and Barbara Beeton for invaluable help and improvements to this article.

References


\bigbreak

\begin{tabular}{l}
\textcopyright{} Ivan R. Pagnossin \\
ivan dot pagnossin (at) gmail.
\end{tabular}

Boxes and more boxes