The labyrinth package

Francesco Zigliotto

April 12, 2014

The labyrinth package provides a code and an environment for typesetting simple labyrinths with \LaTeX and generating an automatic or manual solution path.

CONTENTS

1 A code for writing labyrinths 2
   1.1 Marking horizontal lines 2
   1.2 Marking vertical lines 2
   1.3 Merging horizontal and vertical marks 2
   1.4 Simplifying the code 3
2 The \texttt{labyrinth} environment 3
   2.1 Options of the \texttt{labyrinth} environment 4
   2.2 Adding elements to the labyrinth 4
   2.3 Replacing “+”, “.” and “*” characters 5
3 Typesetting the solution of the labyrinth 5
   3.1 The \texttt{\labyrinthsolution} macro 5
   3.2 Options of the \texttt{\labyrinthsolution} command 5
   3.3 Automatic solution of the labyrinth 6
4 Example 7
Index 7

Figure 1: A labyrinth example (the code is shown at page 7).
The key-point in typesetting a labyrinth with \LaTeX{} is finding a way to describe it by a set of characters. The problem can be solved by including the labyrinth in an ideal $m \times n$ frame, divided into $m \cdot n$ unit squares. The grid is then split into $m$ horizontal sections, of $n$ unit squares each, as reported in Figure 2.

### 1 Marking horizontal lines

First, we have to differentiate between the horizontal and vertical lines of the labyrinth. Let’s focus on horizontal lines. For each unit square, you have to indicate either its bottom line belongs to the labyrinth or not. Accordingly, you will type either “+” or “-”. For example, with reference to Figure 2, the first four unit squares of the second section have their bottom lines included in the labyrinth, while the last one has not. The corresponding description is “++++-”.

### 1.2 Marking vertical lines

Each horizontal section contains $n$ unit squares and thus $(n + 1)$ vertical sides. Similarly to horizontal lines, each vertical side is marked by a “+” if it is a part of the labyrinth and by a “-” if it is not the case. For example, in Figure 2, the vertical lines in the fourth section are marked as “+-++++”.

### 1.3 Merging horizontal and vertical marks

Now we have to arrange the horizontal and vertical descriptions into a unique code. For each section, first we write $\backslash v$, followed by the description of the vertical lines and then $\backslash h$, followed by the description of horizontal lines.
Be careful to keep the correct sequence (\v leading \h), because every \v command also increases the labyrinth section number which we are referring to.

Very often, the first section has no vertical lines (while horizontal lines are quite common). If this is the case, we may start the code with \h, skipping the \v command. Out of the first section, any other one without vertical sides belonging to the labyrinth calls for the \v command.

With the above rules, the code of the labyrinth of Figure 2 is written as follows:

| \h +++++ | \v -----+ \h ++++ | \v +----- | \h ++++++ | \v ++++++ | \v ++++++ | \v +++-+ |
| % 1st section | % 2nd section | % 3rd section | % 4th section | % 5th section | % 6th section |

1.4 Simplifying the code

The labyrinth code can be simplified, by using the rules listed below:

- all “-” characters at the end of each section can be omitted, for both horizontal and vertical marks;
- the \h command can be omitted for any section void of horizontal lines;
- if there are more than three consecutive “+” or “-” characters, you can use the following syntax:
  
  \v *\{n\}+ or *\{(n)\}+

where \(n\) is the number of “+” or “-”.

For example, the code of Section 1.3 can be simplified like this:

| \h *5+ | \v *5+- | \h *4+ | \v ++++++ | \v +*-4+ | \v ++++++ | \v +++-+ |
| % 1st section | % 2nd section | % 3rd section | % 4th section | % 5th section | % 6th section |

2 The labyrinth Environment

To typeset a labyrinth with the labyrinth package, you can use the labyrinth environment:
\begin{labyrinth}\{\textit{options}\}\{\textit{width}\}\{\textit{height}\}
\end{labyrinth}

where:

- \textit{width} is the number of columns ($n$) of the ideal grid described in Section 1 (see Figure 2);
- \textit{height} has to be set to ($m-1$), where $m$ is the number of horizontal sections, since only the bottom lines of the first horizontal section belong to the labyrinth (see Figure 2);
- \textit{options} are optional parameters in the form key=value of the \texttt{xkeyval} package (see Subsection 2.1);
- \textit{labyrinth code} is the code using the \texttt{\v} and \texttt{\h} commands, as described in Section 1.

2.1 Options of the \texttt{labyrinth} environment

There are some options for the labyrinths, which you can put either in the optional argument of the \texttt{labyrinth} environment or in the argument of the \texttt{labyrinthset} command, which should be placed outside the labyrinth environment. It that case, the options work for all the labyrinths from then on.

\labyrinthset{\{\textit{options}\}}

The possible \textit{options}, in the form key=value, are:

- \texttt{unit} (default: 11pt) sets the width of the side of every unit square of the ideal grid (see).
- \texttt{thickness} sets the thickness of the lines of the labyrinth.
- \texttt{centered} (values: true/false) centres horizontally the labyrinth. It also leaves extra space before and after.

2.2 Adding elements to the labyrinth

The content of the \texttt{labyrinth} environment is internally put inside a \texttt{picture} environment, so that if we need to add to the labyrinth oblique lines, symbols or so we can do it with the usual \LaTeX{} command \texttt{\put, \line}...

To put symbols inside the labyrinth, we can use the \texttt{\putsymbol} command, similar to \texttt{\put}, except for it centres its argument horizontally inside the ideal square:
2.3 Replacing “+”, “-” and “*” characters

Of course, inside the \labyrinth environment, we can’t use the characters “+”, “-” and “*” outside the foreseen position, which are substituted by the \plus, \minus and \ast commands, respectively.

3 Typesetting the solution of the labyrinth

3.1 The \labyrinthsolution macro

If we need to typeset the solution of a labyrinth, we can use the \labyrinthsolution command with the following syntax:

\labyrinthsolution\{(options)\}\{(x,y)\}\{(solution code)\}

where:

- \{options\} are the optional parameters of the command in the form key=value of the xkeyval package (see the next Subsection 3.2).

- \{x,y\} are the horizontal and vertical coordinates of the starting point of the solution. Please note that these coordinates are automatically increased by half a \unitlength (for more details, see the options hcorr and vcorr in the Subsection 3.2).

- \{solution code\} is a sequence of characters of the set \{u,l,d,r\}. From the starting point \((x,y)\) we describe each segment of the solution path by indicating the direction: up \(u\), left \(l\), down \(d\) or right \(r\). Each step is one \unitlength long.

Please note that:

- the \labyrinthsolution command should be put inside the \labyrinth environment;

- there can be more than one \labyrinthsolution command inside the same labyrinth (e.g. for multiple solutions).

3.2 Options of the \labyrinthsolution command

As the labyrinth options and the \labyrinthset macro (Subsection 2.1), you can put the solution options either in the optional argument of the \labyrinthsolution command or in the argument of the \solutionset command, that should be placed outside the labyrinth environment and that works for all the labyrinth solutions from then on.
The labyrinth package: Francesco Zigliotto

\solutionset{(options)}

The possible \textit{(options)} are:

\textbf{hidden} \hspace{1cm} (values: true/false, default: false) hides (true) or shows (false) the solution to the labyrinth.

\textbf{thicklines} \hspace{1cm} (values: true/false, default: true) sets the lines of the solution thick (true) or thin (false);

\textbf{up} \hspace{1cm} (default: \texttt{\line(0,1){1}}) defines the symbol that indicates a step up in the solution path (letter \texttt{u});

\textbf{left} \hspace{1cm} (default: \texttt{\line(-1,0){1}}) as above, for the left step (letter \texttt{l});

\textbf{down} \hspace{1cm} (default: \texttt{\line(0,-1){1}}) as above, for the down step (letter \texttt{d});

\textbf{right} \hspace{1cm} (default: \texttt{\line(1,0){1}}) as above, for the right step (letter \texttt{r});

\textbf{hcorr} \hspace{1cm} (default: \texttt{0.5\unitlength}) sets the increment of the horizontal coordinate (it moves horizontally all the solution route);

\textbf{vcorr} \hspace{1cm} (default: \texttt{0.5\unitlength}) sets the increment of the vertical coordinate (it moves vertically all the solution route);

\textbf{font} \hspace{1cm} (default: \texttt{\color{red}if \color} is defined) select the font (mainly the colour) of the labyrinth solution.

3.3 \textit{Automatic solution of the labyrinth}

The package also provides the macro \texttt{\autosolution} that finds and draws automatically one of the labyrinth solutions (if any, of course):

\texttt{\autosolution{(options)}((x_A,y_A))((x_B,y_B))({(first direction)})}

where:

- \textit{(options)} are the same optional parameters of the \texttt{\labyrinth-solution} command described in Section \ref{sec:labyrinth};

- \texttt{(x_A,y_A)} and \texttt{(x_B,y_B)} are the horizontal and vertical coordinates of the starting point (A) and of the arrival point (B) of the solution path. Please note that the coordinates of both points are automatically increased by half a \texttt{\unitlength} (for more details, see options \texttt{hcorr} and \texttt{vcorr} in the Subsection \ref{sec:labyrinth}).

- \textit{(first direction)} is one character of the set \{u,d,l,r\} (up, down, left, right - respectively) that indicates the direction of the first step of the solution path.

The \texttt{\autosolution} command also defines \texttt{\solutionpath}, which generates the string of direction-characters that defines the solution path.
Here you can see the code used for the labyrinth of Figure 1:

```latex
\begin{labyrinth}[unit=9pt]{19}{20}
\putsymbol(8,\minus1){\Large$\Uparrow$}
\putsymbol(10,20){\Large$\Uparrow$}
\h*9+---*7+
\v+*5-+--+*4+--+
\v+*4-+--*3+--+
\v+*4-+--*3+--+
\v+*3-+--*2++--
\v+*2--++*1-++
\v+*1-++*0+++
\end{labyrinth}
```

The labyrinth package: Francesco Zigliotto
The labyrinth package: Francesco Zigliotto

A
\ast, 5
\autosolution, 6

C
centered (opt.), 4
\color, 6

D
d, 5, 6
down, 6

E
Environment
labyrinth, 3, 5
labyrinth, 1, 3–5
picture, 4

F
font, 6

H\h, 2–4
hcorr, 6
hcorr (opt.), 5, 6
hidden (opt.), 6

L\l, 5, 6
labyrinth (env.), 3, 5
\labyrinthsolution, 5
labyrinth (pack.), 1, 3
labyrinth (env.), 1, 3–5
\labyrinthsolution, 6
\labyrinthset, 4, 5
\labyrinthsolution, 1, 5
left, 6
\line, 4

M\minus, 5

O
Option
centered, 4
hcorr, 5, 6
hidden, 6
thickness, 4
unit, 4
vcorr, 5, 6

P
Package
labyrinth, 1, 3
\xkeyval, 4, 5
picture (env.), 4
\plus, 5
\put, 4
\putsymbol, 4

R\r, 5, 6
right, 6

S\solutionpath, 6
\solutionset, 5

T
\solutionpath, 6
\solutionset, 5

\thicklines, 6
\thickness (opt.), 4

U\u, 5, 6
\unit (opt.), 4
\unitlength, 5, 6
up, 6

V\v, 2–4
\vcorr, 6
\vcorr (opt.), 5, 6

X\xkeyval (pack.), 4, 5