





Figure 2: The labyrinth grid split into horizontal sections. (The line thickness has been incremented to differentiate the labyrinth lines from the grid).

## 1 A CODE FOR WRITING LABYRINTHS

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  $m$  by  $n$  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.

### 1.1 *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 `\v`, followed by the description of the vertical lines and *then* `\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 +++++ % 1st section
\v -----+ \h +++++ % 2nd section
\v +++-+++ \h ----- % 3rd section
\v ++++++ \h -+---- % 4th section
\v +---+-+ \h ----+ % 5th section
\v +---+-+ \h +++++ % 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:

---

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

```

---

## 2 THE Labyrinth ENVIRONMENT

To typeset a labyrinth with the labyrinth package, you can use the `labyrinth` environment:

---

```
\begin{labyrinth}[options]{width}{height}
labyrinth code
\end{labyrinth}
```

---

where:

- $\langle width \rangle$  is the number of columns ( $n$ ) of the ideal grid described in Section 1 (see Figure 2);
- $\langle height \rangle$  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);
- $\langle options \rangle$  are optional parameters in the form `key=value` of the `xkeyval` package (see Subsection 2.1);
- $\langle labyrinth code \rangle$  is the code using the `\v` and `\h` commands, as described in Section 1.

### 2.1 Options of the `\labyrinth` environment

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

---

```
\labyrinthset{options}
```

---

The possible  $\langle options \rangle$ , in the form `key=value`, are:

`unit` (default: 11pt) sets the width of the side of every unit square of the ideal grid (see).

`thickness` sets the thickness of the lines of the labyrinth.

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

To put symbols inside the labyrinth, we can use the `\putsymbol` command, similar to `\put`, except for it centres its argument horizontally inside the ideal square:

---

```
\putsymbol( $\langle h-pos \rangle$ , $\langle v-pos \rangle$ ){ $\langle symbol \rangle$ }
```

---

### 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[ $\langle options \rangle$ ]( $\langle x,y \rangle$ ){ $\langle solution code \rangle$ }
```

---

where:

- $\langle options \rangle$  are the optional parameters of the command in the form `key=value` of the `xkeyval` package (see the next Subsection 3.2).
- $\langle x,y \rangle$  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).
- $\langle solution code \rangle$  is a sequence of characters of the set  $\{u,l,d,r\}$ . From the starting point  $(\langle x,y \rangle)$  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.

---

`\solutionset{options}`


---

The possible *options* are:

`hidden` (values: `true/false`, default: `false`) hides (`true`) or shows (`false`) the solution the labyrinth.

`thicklines` (values: `true/false`, default: `true`) sets the lines of the solution thick (`true`) or thin (`false`);

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

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

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

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

`hcorr` (default: `0.5\unitlength`) sets the increment of the horizontal coordinate (it moves horizontally all the solution route);

`vcorr` (default: `0.5\unitlength`) sets the increment of the vertical coordinate (it moves vertically all the solution route);

`font` (default: `\color{red}` if `\color` is defined) select the font (mainly the colour) of the labyrinth solution.

### 3.3 Automatic solution of the labyrinth

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

---

`\autosolution[options](xA,yA)(xB,yB){first direction}`


---

where:

- *options* are the same optional parameters of the `\labyrinth-solution` command described in Section 3.2;
- *x<sub>A</sub>,y<sub>A</sub>* and *x<sub>B</sub>,y<sub>B</sub>* 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 `\unitlength` (for more details, see options `hcorr` and `vcorr` in the Subsection 3.2).
- *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 `\autosolution` command also defines `\solutionpath`, which generates the string of direction-characters that defines the solution path.

## 4 EXAMPLE

Here you can see the code used for the labyrinth of Figure 1:

---

```

\begin{labyrinth}[unit=9pt]{19}{20}
\putsymbol(8,\minus1){\Large$\Uparrow$}
\putsymbol(10,20){\Large$\Uparrow$}
      \h *9+---*7+
\V +*5-+---+---+---+---+ \h -*4+---+---+*4+---+
\V +*4-+---+---+---+---+ \h *4+---+---+---+---+
\V +---+---+---+---+---+ \h -+---+---+---+*4-+---+
\V +---+---+---+---+---+ \h +-+---+---+---+---+---+
\V +---+---+*4-+---+*4+---+ \h -+---+---+*4+---+*6-+
\V +---+---+---+---+---+ \h +---+---+---+---+---+---+
\V +---+---+---+---+---+ \h -+---+---+---+---+---+
\V +---+---+---+---+---+ \h +---+---+---+---+---+
\V +---+---+---+---+---+ \h -+*7-+---+*4-+---+
\V +-+*6+---+---+---+ \h ---+*8-+*4+---+
\V +-+*8+---+---+---+ \h -+---+*{12}-+
\V +-+---+---+---+---+ \h +---*7+---+---+
\V +-+---+---+---+---+ \h -+---+---+---+---+---+
\V +-+---+---+---+---+ \h ---+---+*5-+---+
\V +---+---+---+---+---+ \h -+---+---+---+*4-+
\V +-+---+---+---+---+ \h -+---+---+*4-+---+
\V +---+---+---+*5+---*4+ \h -+---+---+
\V +-+*4+---+---+*6+ \h *6-+---+*5+
\V *4+---+*8-+---+ \h -+---+---+*6+
\V +-+*4-+---+*8-+ \h *7+---+*9+

\autosolution[font=\color{orange}](8,0)(10,19){u}

\labyrinthssolution[
font=\color{MidnightBlue}\footnotesize,
up=\kern2pt$\uparrow$,
left=$\leftarrow$,
down=\kern2pt$\downarrow$,
right=$\rightarrow$,
hcorr=0.1\unitlength,
vcorr=0.3\unitlength](7,1){%
ldllllluulddluuuuuuuruuuurulurrdd%
dlrruuurrrdddluuuldddllrrrdr}
\end{labyrinth}

```

---

## INDEX

## SYMBOLS

-, 1–3, 5

\*, 1, 3, 5

+, 1–3, 5

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
- `\labyrinthssolution`, 5
- labyrinth (*pack.*), 1, 3
- labyrinth (*env.*), 1, 3–5
- `\labyrinthssolution`, 6
- `\labyrinthset`, 4, 5
- `\labyrinthssolution`, 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

- 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