The `multi-sudoku` Package
User Guide

Luis F. Bartolo Alegre
Email: `l.bartolo@campus.lmu.de`

Version 1.00 – 31st July 2025

Contents

1 Introduction

1.1 What is `multi-sudoku`?

The `multi-sudoku` package provides tools for typesetting Sudoku grids of various sizes in LaTeX. Unlike other Sudoku packages which are typically limited to the standard 9×9 layout, this package supports a broad range of grid sizes – from trivial 1×1 puzzles to extended 49×49 Sudokus – that's the limit for now!

Grids are drawn with our `sudoku` environment, which is based on using LaTeX's native `tabular` environment. We include intuitive options to control dimensions, font size, and grid thickness. Entries in the grid are inserted as in a regular table, thus making it simple to create, customise, and fill Sudoku puzzles manually.

1.2 Key features

- Supports various Sudoku grid sizes: 1×1, 2×2, 4×4, 9×9, 16×16, 25×25, 36×36, 49×49.
- Easy manual entry of values using standard tabular syntax.
- Per-grid and global styling options for Sudoku size (`size`), cell side length (`length`), font size (`fontsize`), grid line thickness (`thickness`).
- Lightweight: no external drawing libraries like TikZ required.
- Compatible with plain LaTeX workflows.

1.3 Why another Sudoku package?

Unlike other packages that focus solely on standard Sudoku, `multi-sudoku` was designed from the ground up to support multiple puzzle sizes and offer control over visual layout. This flexibility makes it ideal not just for standard 9×9 Sudokus, but also for educators, researchers, and enthusiasts working with mini-puzzles, non-standard formats, or large-scale variants.

1.4 Requirements

The package relies only on standard LaTeX tools and the following commonly available packages: `pgfkeys`, `array`, `xparse`, `calc`, `kvoptions`, `ifthen`. These are usually bundled in all major LaTeX distributions (e.g., TeX Live, MiKTeX).

1.5 Licensing and author

Author: Luis F. Bartolo Alegre

Email: `l.bartolo@campus.lmu.de`

Version: 1.00 (2025/7/31)

Licence: LaTeX project public license (LPPL), version 1.3c.

2 Installation and loading

2.1 Installation

The `multi-sudoku` package can be installed by copying the `multi-sudoku.sty` file into a directory where LaTeX can find it, like the main folder of your project.

2.2 Loading the package

To use `multi-sudoku`, add the following line to your document preamble:

```
\usepackage{multi-sudoku}
```

The package loads with default options suitable for standard 9×9 Sudoku grids. You can set global options at package load time, the following being the default ones:

```
\usepackage[
    size=9x9,
    length=1.5em,
    thickness=1.5pt,
    fontsize=\small
    ]{multi-sudoku}
```

where:

- `size` specifies the grid size (e.g., `9x9`, `16x16`).
- `length` sets the cell side length.
- `thickness` sets the thickness of grid lines.
- `fontsize` sets the font size inside cells.

3 Basic usage

The `multi-sudoku` package defines the `sudoku` environment, which draws a Sudoku grid of the specified size. The content of the grid is input just like a standard `tabular` environment, with one cell per Sudoku square.

Below is a basic example of a blank 9×9 Sudoku grid, in which the values are entered into the cells as usual in a `tabular`:

```
\begin{sudoku}
    5 &   &   &   & 7 &   &   &   & 9 \nl
    &   & 3 &   &   &   & 1 &   &   \nl
    & 1 &   & 9 &   &   &   & 6 &
    \NL
    &   & 8 &   &   & 6 &   & 3 &   \nl
    & 9 &   &   & 5 &   &   & 1 &   \nl
    6 &   &   &   &   & 1 &   &   &
    \NL
    & 2 &   & 3 &   &   & 6 &   &   \nl
    &   &   &   & 2 &   &   & 7 &   \nl
    1 &   &   &   &   & 8 &   &   & 4
\end{sudoku}
```

And here is the compiled result:

5				7				9
		3				1		
	1		9				6	
		8			6		3	
	9			5			1	
6					1			
	2		3			6		
				2			7	
1					8			4

Note: The `\nl` command inserts a new row with a standard horizontal line above it (used for regular cell boundaries). In contrast, the `\NL` command inserts a new row preceded by a thicker horizontal line – typically used to separate Sudoku subgrids. The thickness of this line is controlled by the `thickness` option, which can be set globally or locally (see section 5).

4 Supported sizes

The `multi-sudoku` package supports the following Sudoku grid sizes, where each size corresponds to a Sudoku composed of $n \times n$ cells arranged as k^2 blocks of size $k \times k$:

size option	**description**
1x1	Trivial Sudoku: a single cell
2x2	Four 1×1 subgrids (pseudo-Sudoku, mostly for demonstration)
4x4	Four 2×2 subgrids
9x9	Standard Sudoku: nine 3×3 subgrids (default size)
16x16	Sixteen 4×4 subgrids
25x25	Twenty-five 5×5 subgrids
36x36	Thirty-six 6×6 subgrids
49x49	Forty-nine 7×7 subgrids

Attempting to use unsupported sizes will result in an error.

Note: The column layouts for each supported grid size have been manually defined, which limits the package to a fixed set of predefined sizes. If you have ideas for how to generate these column arrangements recursively – so that arbitrary grid sizes could be supported – feel free to get in touch.

5 Options and customisation

The package offers several options that control the appearance and layout of Sudoku grids. These can be set either globally when loading the package or locally as environment options.

5.1 Global options

Set these options when loading the package via `\usepackage`:

- `size`
  Defines the size of the Sudoku grid. Defaults to `9x9`. Examples: `4x4`, `16x16`.
- `length`
  Specifies the length of the side of each Sudoku cell. Default is `1.5em`. Accepts any length unit (`em`, `pt`, `cm`, etc.).
- `thickness`
  Sets the thickness of the grid lines (especially the thicker subgrid boundaries). Default is `1.5pt`.
- `fontsize`
  Sets the font size inside each cell. Default is `\small`. Can be any valid font size command (e.g., `\footnotesize`, `\normalsize`).

For example, let us load the following in the preamble:

```
\usepackage[
size=4x4,
length=3.25,
thickness=1pt,
fontsize=\large
]{multi-sudoku}
```

Moreover, let us, within that very file, lead the following command:

```
\begin{sudoku}
    $C_{111}$ & $C_{121}$ & $C_{134}$ & $C_{144}$
    \nl
    $C_{211}$ & $C_{221}$ & $C_{234}$ & $C_{244}$
    \NL
    $C_{312}$ & $C_{322}$ & $C_{333}$ & $C_{343}$
    \nl
    $C_{412}$ & $C_{422}$ & $C_{433}$ & $C_{443}$
\end{sudoku}
```

Thus, we will obtain the following result:

C_{111}	C_{121}	C_{134}	C_{144}
C_{211}	C_{221}	C_{234}	C_{244}
C_{312}	C_{322}	C_{333}	C_{343}
C_{412}	C_{422}	C_{433}	C_{443}

5.2 Local options

You can override global options for individual Sudoku environments by passing key-value options as an optional argument to the `sudoku` environment. Below are some examples assuming the default options.

Example 1: The trivial 1×1 Sudoku with otherwise default settings – and solved!

Command:

```
\begin{sudoku}[size=1x1]
    1
\end{sudoku}
```

Output:

1

Example 2: A 2×2 pseudo-Sudoku with otherwise default settings.

Command:

```
\begin{sudoku}[size=2x2]
    1 & 4
    \NL
    2  & 3
\end{sudoku}
```

Output:

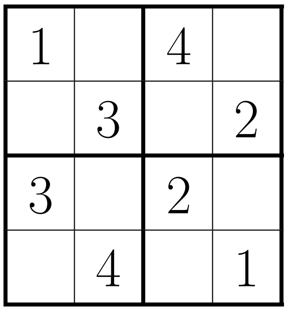
1	4
2	3

Example 3: A 4×4 Sudoku with `\LARGE` font size.

Command:

```
\begin{sudoku}[size=4x4, fontsize=\LARGE]
    1 &   & 4 &  \nl
    & 3 &   & 2
    \NL
    3 &   & 2 &  \nl
    & 4 &   & 1
\end{sudoku}
```

Output:



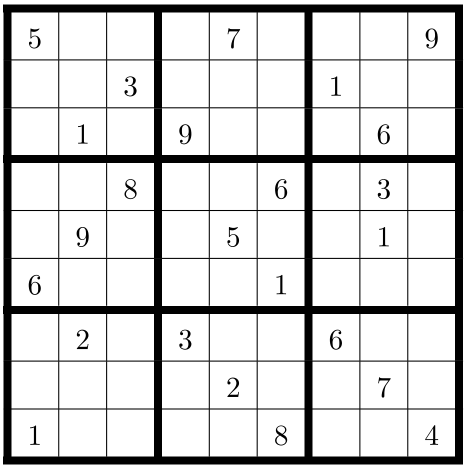
1		4	
	3		2
3		2	
	4		1

Example 4: A standard 9×9 Sudoku with grid lines set to 3pt thickness. The `size` option can be omitted unless a different default was specified globally.

Command:

```
\begin{sudoku}[size=9x9, thickness=3pt]
    5 &   &   &   & 7 &   &   &   & 9 \nl
    &   & 3 &   &   &   & 1 &   &   \nl
    & 1 &   & 9 &   &   &   & 6 &
    \NL
    &   & 8 &   &   & 6 &   & 3 &   \nl
    & 9 &   &   & 5 &   &   & 1 &   \nl
    6 &   &   &   &   & 1 &   &   &
    \NL
    & 2 &   & 3 &   &   & 6 &   &   \nl
    &   &   &   & 2 &   &   & 7 &   \nl
    1 &   &   &   &   & 8 &   &   & 4
\end{sudoku}
```

Output:



5				7				9
		3				1		
	1		9				6	
		8			6		3	
	9			5			1	
6					1			
	2		3			6		
				2			7	
1					8			4

Example 5: A 16×16 Sudoku with font size set to `\scriptsize` and cell side length set to 1em.

Command:

```
\begin{sudoku}[size=16x16, length=1em, fontsize=\scriptsize]
    ... % grid contents
\end{sudoku}
```

Output:

A						7							3		
		2			5				C						6
			F	1							D			8	
9						3		B							
	3			6								1			
			7				4		5					2	
		D						2			3				
B					1								F		
		9						D		4					
4						1			E						C
	5			3										B	
			C				9					7			
7										5					
			1				6						8		
	6									2					
			8					4							1

Example 6: A 25×25 Sudoku with cell side length set to 1.25em.

Command:

```
\begin{sudoku}[size=25x25, length=1.25em]
    ... % grid contents
\end{sudoku}
```

Output:

1		4	9			7		A					3	5				5			8		6	
		2		8	5		1		C	6		8		9	6		4			7				
			F	1	3			3			D			8			7					2		1
9	4			7		3		B			2		1				5				6	4		
	3			6	9	7				1		1				2			8				9	
		6	7	4			4		5				7	2			3				1			
		D		8	7			2		1	3			6			9	5					7	
B				4	1						6		F			8				3			2	
		9	5			3		D		4		2		8		7						9		6
4			6	2		1	8		E				9	4	C			7		1		5		
	5		9	3	6			7						B	1			4			3			
			C			8	9				5	7					2	1						4
7	1					4			3	5		6								9			8	
			1	8			6					3	8		7				2				1	
	6			7					4	2				9		3			5					
	8		8					4		7					1			6			7			3
			3				7		9			1								4	6			
					2			1						5			3					8		9
						6				8			4						7	9				
					5							7						4		1			6	
						5				9				6			8				2			
		7					1		3				2						5			8	4	
	1							8													9			9
			4					7		6						1							3	
				9					5								6							

Example 7: A 36×36 Sudoku with cell side length set to 1em and font size set to `\scriptsize`.

Command:

```
\begin{sudoku}[size=36x36, length=1em, fontsize=\scriptsize]
    ... % grid contents
\end{sudoku}
```

Output:

4	9		B	1				7			5			6			A							3			2								
	6			2		D			C	8			7		9					1															
3			F	4	5							D				8													7						
9					3		1		B		4			7				6																	
	3	5		6				2				9	1									8							4						
			7				8	4		5								9				3													
		D		7					2	6		3				5											4								
B					1	7							F		2				9																
	8	9							D		4				5											1									
4				6		1				E		7				8							3												
	7	6	7					4	2	5								1			8														
			7				9	4		5		1				7										6									
	5	8		3		2		6				4		B							7							9							
			C		4		9				3	7			8			1				2							5						
7				9						5		1							3											4					
		2	1				6						8		7					9															
			7			5		4		5					3								8						1						
			7		1			4		5	6								7					9											
	5			3		7					1			B		4				6					2										
			C			8	9		6			7					1														5				
8										5	7						9			1				3							4				
			1		6		6					2	8							7		4													
		7						4		5			3							8			1								9				
			7					4		5					7		2					6					8							
	5			3		2								B			7					4				8									
			C	1			9			7		7						6						1				5							
7						4				5					9				3							7				1					
			1				6		8				8					2				5													
	7			7				4		5					4					7				3						9					
			7					4		5		3										7			2					5				
	5			3			1			8				B					6						4				7						
			C				9				4	7						1		5									8						
9										5							3			7				2							4				
			1				6				8							9								5								
			7					4		5			1			7					8														

Example 8: A 49×49 Sudoku with cell side length set to .75ememem and font size set to `\tiny`.

Command:

```
\begin{sudoku}[size=49x49, length=.75em, fontsize=\tiny]%
    ... % grid contents
\end{sudoku}
```

Output:

9		1			7		5					2		6			8			3				9		4		1			7		5		8		2			6		3		7		1		9
7			3		8			4			1		5						7		6		1		8					4		9		2		5				3		7		6		1		8
	6			9		2			5		4			7		3			8			1			6		9			2			7		1		8			3		5		7		9		
8		4			6		1			7	9		5			3		2			8		6			1		4			9			5			7		3		9		6			1		
	2			7			3		6	8		4			9		1			5			7		2		9		3			6			8		4		1			7		5			9	
1			5			7		2		8			3			6		4			9		1			5		7		8			3			6		4			9		1			5		7
	3		8			5		7			6			9			1		3			4		7		5		6			9			3			4			7		5		6		9		1
6		1			4			2			7		8			5			3		1		4		9			2			7		8			5			3		1		4		9			2
	7			3		1			9		5			2		6			8			4		3		1			5			2		6			8			4		3		1		9		5
2			9		5			4			8		1			7			6			9		5			4		3			8		1			7			6			9		5			4
		8			6		7			1		9			3		5			2			8		6		7			1			3		5			2			8		6		7			
5		9		7			4		2				8			1		3			5		9		7			4		2			6			8			1		3			5		9		7
	1			8			9			4			7			5			2			6		1			8			9			4			7			5			2			6		1	
3			2		9			5			7			1			4				3			2		9			5			7			1			4			8		3			2		
	4			7		6			1		9			3			5			2			4		7		6			1		9			3			5			2			4		7		6
9		5			3			8		7			4			1			6		9		5			3			8		7			4			1			6		9		5				8
	8		7			4			1		6			9			5			3			8		7			4			1		6			9			5			3			8		7	
1			5			2		9			6			3			8			4			7		1			5			2		9			6			3			8			4			1
	9			4		7			3			8			5			2			6			1		9			4			7			3			8			5			2				
4		7			1		5			9			6			3			8			4		7			1		5			9			6			3			8			4		7		
	2		6			3			8		4			7			1			5			9		2			6			3			8		4			7			1			5			9
5		1		9			7			2			4			8			3			6		5		1		9			7			2			4			8			3			6		5
	3			6		8			9			5			2			7			4		3			6			8			1		9			5			2			7			4		
2			5		7				3			9			6			1			8		2			5			7			4			3			9			6			1			8	
	8			1			5			7			2			4			9			3			8			1			5			7			2			4			9			3		
7			4			9			6			1			8			3			5			7			4			9			6			1			8			3			5			7
	5			3			1			9			7			2			6			4			5			3			1			9			7			2			6			4		
9		6			4			8			5			1			7			3			9		6			4			8			5			1			7			3			9		6
	1		7			2			9			4			3			8			6			1			7			2			9			4			3			8			6			1
4			9		3			5			7			1			8			2			4			9		3			5			7			1			8			2			4		
	7			5			8			3			6			9			1			4			7			5			8			3			6			9			1			4		
3		2			9			1			5			7			4			6			3		2			9			1			5			7			4			6			3		2
	6			8			4			7			1			5			9			2			6			8			4			7			1			5			9			2		
8		9			1			6			3			5			7			8		9			1			6			3			5			7			8		9			1			6
	4		3			7			2			9			8			1			5			4		3			7			2			9			8			1			5			4	
1			5		8			9			6			4			7			3			1			5		8			9			6			4			7			3			1		
	9			2			7			5			1			8			4			6			9			2			7			5			1			8			4			6		
2		6			4			8			3			7			9			1			2		6			4			8			3			7			9			1			2		6
	5			3			1			9			6			4			7			5			3			1			9			6			4			7			5			3		
7			8			2			5			9			1			3			6			7			8			2			5			9			1			3			6			7
	1			9			4			6			3			7			8			2			1			9			4			6			3			7			8			2		
4		3			7			1			5			8			9			2			4		3			7			1			5			8			9			2			4		3
	6			8			2			7			4			1			5			9			6			8			2			7			4			1			5			9		
5			9			3			6			1			8			7			4			5			9			3			6			1			8			7			4			
	2		4			7			9			5			3			8			1			2			4			7			9			5			3			8			1			
9		7			1			8			4			6			5			2			9		7			1			8			4			6			5			2			9		7
	3			6			9			1			7			4			8			3			6			9			1			7			4			8			3			6		
1			5			2			8			3			9			7			6			1			5			2			8			3			9			7			6			1
	8			4			7			9			6			5			1			2			8			4			7			9			6			5			1			2		

6 Compatibility

- The package relies on `pgfkeys`, `array`, `xparse`, `kvoptions`, and `ifthen`, so these must be installed and up to date.
- It may be incompatible with other Sudoku-related packages that redefine tabular environments or grid drawing.
- Since the package uses fixed tabular layouts for each supported size, unsupported sizes will trigger an error.
- The column types `S` and `T` are predefined by the package: `S` corresponds to standard cells, while `T` represents thick vertical borders. To ensure proper grid rendering, avoid redefining these column types.
- Cell content should be plain text or use simple text-formatting commands. If mathematical expressions are needed, they should be enclosed in standard math environments such as `$...$` or `\(...\)`.

7 Closing

Please report bugs or suggestions to `l.bartolo@campus.lmu.de`.

Happy puzzling!