

The **sdiv** Package

Drawing Successive Division Diagrams in L^AT_EX

Joao Alberto Coelho

japbcoelho@proton.me

Version 1.0.0
2025-12-05

Abstract

The **sdiv** package provides a simple and elegant way to draw successive division diagrams in L^AT_EX. These diagrams are commonly used in mathematics education for teaching base conversion, division algorithms, and number theory concepts. The package offers customizable options for highlighting, arrows, and circle sizes.

Contents

1	Introduction	1
1.1	Requirements	1
1.2	Installation	1
1.2.1	Via CTAN (when available)	1
1.2.2	Manual Installation	1
2	Basic Usage	2
2.1	Loading the Package	2
2.2	The \sdiv Command	2
2.3	Simple Example	2
3	Options	2
3.1	Circle Highlighting	2
3.2	Arrow	2
3.3	Underline Highlighting	3
3.4	Combining Options	3
3.5	Custom Circle Size	3
3.6	Comparison of Circle Sizes	4
4	Applications	4
4.1	Base Conversion	4
4.1.1	Binary (Base 2)	4
4.1.2	Octal (Base 8)	4
4.1.3	Hexadecimal (Base 16)	5
4.2	Teaching Division	5
5	Input Validation	5
6	Layout and Spacing	5
6.1	Fixed Gap	5
6.2	Dynamic Column Width	6

7 Examples	6
7.1 Edge Cases	6
7.1.1 Dividend Less Than Divisor	6
7.1.2 Dividend Equal to Divisor	6
7.1.3 Zero Dividend	6
7.2 Large Numbers	6
8 Known Limitations	7
8.1 LuaLaTeX Only	7
8.2 Circle Overlap with Large Inner Sep	7
8.3 Font Dependency	7
9 Future Enhancements	7
10 Technical Details	7
10.1 Implementation	7
10.2 Algorithm	7
11 License	8
12 Contact	8
13 Version History	8

1 Introduction

Successive division is a fundamental algorithm used primarily for converting numbers between different bases. The process involves repeatedly dividing a number by a base and recording the remainders. When read in reverse order (from bottom to top), these remainders form the digits of the number in the new base.

This package provides the `\sdiv` command to typeset these diagrams in a clean, professional format with automatic layout and spacing.

1.1 Requirements

- **LuaLaTeX:** This package requires LuaLaTeX for execution. It will not work with pdfLaTeX or XeLaTeX.
- **TikZ:** The TikZ package is automatically loaded and used for drawing.

1.2 Installation

1.2.1 Via CTAN (when available)

Once the package is available on CTAN, you can install it using your TeX distribution's package manager:

```
tlmgr install sdiv
```

1.2.2 Manual Installation

Download `sdiv.sty` and place it either:

- In your local `texmf` tree (run `texhash` afterward)
- In the same directory as your `.tex` file

2 Basic Usage

2.1 Loading the Package

```
\usepackage{sdiv}
```

2.2 The \sdiv Command

The main command provided by this package is:

```
\sdiv[options]{dividend}{divisor}
```

options (Optional) Comma-separated list of options

dividend Non-negative integer to be divided

divisor Positive integer (the base for conversion)

2.3 Simple Example

```
\[\sdiv{97}{5}\]
```

produces:

$$\begin{array}{r} 97 \\ 2 \end{array} \left| \begin{array}{r} 5 \\ 19 \\ 4 \end{array} \right. \begin{array}{r} 5 \\ 3 \end{array}$$

3 Options

3.1 Circle Highlighting

Use `circle` or `highlight` to draw circles around the remainders and the final quotient:

```
\[\sdiv[circle]{97}{5}\]
```

$$\begin{array}{r} 97 \\ \textcircled{2} \end{array} \left| \begin{array}{r} 5 \\ 19 \\ \textcircled{4} \end{array} \right. \begin{array}{r} 5 \\ \textcircled{3} \end{array}$$

3.2 Arrow

Use `arrow` to add an arrow showing the reading direction (useful for base conversion):

```
\[\sdiv[arrow]{97}{5}\]
```

$$\begin{array}{r} 97 \\ 2 \end{array} \left| \begin{array}{r} 5 \\ 19 \\ 4 \end{array} \right. \begin{array}{r} 5 \\ 3 \end{array}$$


3.3 Underline Highlighting

Use `underline` to underline the remainders and the final quotient using the `ulem` package:

```
\[\sdiv[underline]{97}{5}\]
```

$$\begin{array}{r} 97 \quad | \quad 5 \\ 2 \quad 19 \quad | \quad 5 \\ \underline{4} \quad \underline{3} \end{array}$$

This provides an alternative to circles for highlighting important values.

3.4 Combining Options

Options can be combined using commas:

```
\[\sdiv[circle,arrow]{97}{5}\]
\[\sdiv[underline,arrow]{97}{5}\]
\[\sdiv[circle,underline,arrow]{97}{5}\]
```

$$\begin{array}{r} 97 \quad | \quad 5 \\ (2) \quad 19 \quad | \quad 5 \\ \nwarrow \quad (4)(3) \\ \underline{4} \quad \underline{3} \end{array}$$

$$\begin{array}{r} 97 \quad | \quad 5 \\ 2 \quad 19 \quad | \quad 5 \\ \nwarrow \\ \underline{4} \quad \underline{3} \end{array}$$

$$\begin{array}{r} 97 \quad | \quad 5 \\ (2) \quad 19 \quad | \quad 5 \\ \nwarrow \quad (4)(3) \\ \underline{4} \quad \underline{3} \end{array}$$

3.5 Custom Circle Size

You can specify a custom inner sep value (in em units) for the circles by providing a number:

```
\[\sdiv[0.5,arrow]{97}{5}\]
```

$$\begin{array}{r} 97 \quad | \quad 5 \\ (2) \quad 19 \quad | \quad 5 \\ \nwarrow \quad (4)(3) \\ \underline{4} \quad \underline{3} \end{array}$$

When a numeric value is specified, circles are automatically enabled. Values typically range from 0.05 (tight) to 1.0 (large). The default is 0.25 when using the `circle` option without specifying a value.

3.6 Comparison of Circle Sizes

Inner Sep	Result
0.05	$\begin{array}{r} 42 \mid 5 \\ \textcircled{2} \ 8 \mid 5 \\ \textcircled{3} \textcircled{1} \end{array}$
0.25 (default)	$\begin{array}{r} 42 \mid 5 \\ \textcircled{2} \ 8 \mid 5 \\ \textcircled{3} \textcircled{1} \end{array}$
0.3	$\begin{array}{r} 42 \mid 5 \\ \textcircled{2} \ 8 \mid 5 \\ \textcircled{3} \textcircled{1} \end{array}$
0.5	$\begin{array}{r} 42 \mid 5 \\ \textcircled{2} \ 8 \mid 5 \\ \textcircled{3} \textcircled{1} \end{array}$

4 Applications

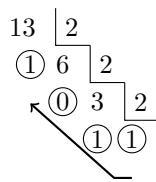
4.1 Base Conversion

The primary use case is converting numbers between bases. The remainders, read from bottom to top, give the digits in the new base.

4.1.1 Binary (Base 2)

Converting 13 to binary:

```
\[\sdiv[circle,arrow]{13}{2}\]
```

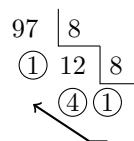


Reading from bottom to top: $13_{10} = 1101_2$

4.1.2 Octal (Base 8)

Converting 97 to octal:

```
\[\sdiv[circle,arrow]{97}{8}\]
```

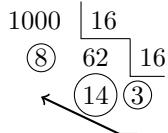


Result: $97_{10} = 141_8$

4.1.3 Hexadecimal (Base 16)

Converting 1000 to hexadecimal:

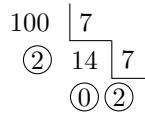
```
\[\sdiv[circle,arrow]{1000}{16}\]
```



Result: $1000_{10} = 3E8_{16}$ (where remainder 14 = E)

4.2 Teaching Division

The diagrams can also be used to visualize the division algorithm in general mathematics courses:



This shows that $100 \div 7 = 14$ remainder 2.

5 Input Validation

The package validates all inputs and produces clear error messages:

Non-integer inputs Error: “Dividend must be an integer”

Zero divisor Error: “Divisor cannot be zero”

Negative dividend Error: “Dividend must be non-negative”

Non-positive divisor Error: “Divisor must be positive”

6 Layout and Spacing

The package uses intelligent spacing to ensure:

- Numbers remain at a fixed distance from vertical bars
- Circles do not overlap with numbers or bars
- Columns are properly aligned
- Large numbers are handled gracefully

6.1 Fixed Gap

All numbers are positioned at a fixed distance (0.2em) from their adjacent vertical bars, regardless of circle size. This ensures consistent layout.

6.2 Dynamic Column Width

Column widths adjust automatically based on:

- The width of numbers in that column
- The size of circles (if present)
- Padding to prevent overlap

7 Examples

7.1 Edge Cases

7.1.1 Dividend Less Than Divisor

```
\[\sdiv[circle]{3}{10}\]
```

$$\begin{array}{r} 3 \longdiv{10} \\ (3)0 \end{array}$$

7.1.2 Dividend Equal to Divisor

```
\[\sdiv[circle]{7}{7}\]
```

$$\begin{array}{r} 7 \longdiv{7} \\ (0)1 \end{array}$$

7.1.3 Zero Dividend

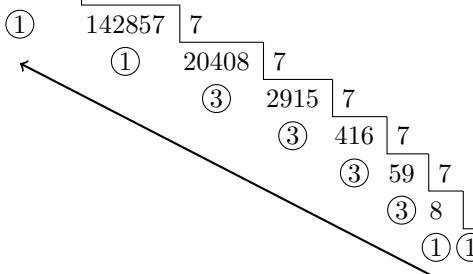
```
\[\sdiv[circle]{0}{5}\]
```

$$\begin{array}{r} 0 \longdiv{5} \\ (0)0 \end{array}$$

7.2 Large Numbers

The package handles large numbers well:

```
\[\sdiv[circle,arrow]{1000000}{7}\]
```

$$\begin{array}{r} 1000000 \longdiv{7} \\ (1) 142857 \longdiv{7} \\ \quad (1) 20408 \longdiv{7} \\ \quad \quad (3) 2915 \longdiv{7} \\ \quad \quad \quad (3) 416 \longdiv{7} \\ \quad \quad \quad \quad (3) 59 \longdiv{7} \\ \quad \quad \quad \quad \quad (3) 8 \longdiv{7} \\ \quad \quad \quad \quad \quad \quad (1)1 \end{array}$$


8 Known Limitations

8.1 LuaLaTeX Only

This package uses Lua code for calculations and layout. It requires LuaLaTeX and will not work with pdfLaTeX or XeLaTeX.

8.2 Circle Overlap with Large Inner Sep

When using very large custom inner sep values (greater than approximately 2.0), the circles may overlap with the vertical bars. This is a known issue that will be addressed in a future version.

Workaround: Use moderate inner sep values (less than 1.0) for best results. The default value of 0.25 is suitable for most use cases.

8.3 Font Dependency

The package uses an approximate character width of 0.55em. This works well for most fonts but may need adjustment for very narrow or very wide fonts. This will be made customizable in a future version.

9 Future Enhancements

Potential improvements for future versions include:

- Better handling of very large inner sep values
- Customizable character width constant
- Optional vertical layout (instead of horizontal)
- Color customization for circles and arrows
- Alternative arrow styles
- Support for custom fonts

10 Technical Details

10.1 Implementation

The package is implemented using:

- **Lua** for the successive division algorithm and layout calculations
- **TikZ** for drawing the diagram elements
- **Dynamic positioning** based on number widths and circle sizes

10.2 Algorithm

The successive division algorithm:

1. Divide the current number by the divisor
2. Record the quotient and remainder
3. Set the current number to the quotient
4. Repeat until the current number is less than the divisor

11 License

This work may be distributed and/or modified under the conditions of the LaTeX Project Public License, either version 1.3c of this license or (at your option) any later version.

The latest version of this license is at:

<http://www.latex-project.org/lppl.txt>

This work has the LPPL maintenance status “maintained”.

12 Contact

Bug reports, suggestions, and contributions are welcome:

Email japbcoelho@proton.me

Author Joao Alberto Coelho

13 Version History

v1.0.0 (2025-12-05) Initial release

For detailed changes, see `CHANGELOG.md`.