The `luanumbers` Package

Document-Wide Decimal Adjustment from a Single Preamble Setup

Parsa Yazdi
Developer and maintainer

Version 0.5.0, 10 June 2026

Summary

`luanumbers` is a LuaLaTeX package that lets an author declare a decimal-formatting policy once in the document preamble and apply it automatically to ordinary decimal literals throughout prose and mathematics. Existing numbers can remain in their natural LaTeX form: they do not normally need to be wrapped individually in formatting commands, moved into special data structures, or prepared with an additional number-formatting package. The Lua implementation uses exact decimal-string arithmetic, preserves integers by default, protects structural LaTeX data, and provides local exclusions for selected sections, figures, tables, diagrams, and other objects. Explicit formatting remains available for ambiguous or protected contexts where the author's intent cannot be inferred safely. Documents using the package must be compiled with LuaLaTeX.

Practical Advantages

- **One-time configuration:** precision, rounding mode, zero padding, and related behavior are declared together in the preamble.
- **Natural source text:** ordinary literals such as `3.1` can be written directly instead of placing every value inside a dedicated formatting command.
- **Efficient document-wide revision:** changing one setup value can update the displayed precision across a long document without editing each occurrence manually.
- **Conservative automation:** integers, labels, references, file names, graphics coordinates, and ambiguous source are preserved according to documented safeguards.
- **Fine-grained control:** authors can exclude one selected object, protect recurring commands or environments, suspend processing temporarily, or request explicit formatting for an individual value.
- **Self-contained workflow:** the package supplies both the global automatic policy and the explicit per-value command, avoiding the need to combine separate number-rounding mechanisms for these tasks.

Contents

1 Purpose and Requirements **2**

2 Installation and Compilation **3**

3 Basic Setup **3**

4 Input and Output **3**
- 4.1 Mathematics . . . 3
- 4.2 Scientific Notation . . . 4
- 4.3 Explicit Formatting . . . 4

5 Configuration Reference **4**
- 5.1 Rounding Modes . . . 4

6 Exact Decimal Arithmetic **4**

7 Local Object Exclusion **4**
- 7.1 One Selected Section . . . 5
- 7.2 One Selected Figure . . . 5
- 7.3 One Selected Table . . . 5

8 Globally Protected Environment Types **5**

9 Protected Commands **6**

10 TikZ, PGFPlots, and Beamer **6**

11 Warnings and Ambiguous Input **7**

12 Manual Suspension **8**

13 Limitations **8**

14 Development and Verification **8**

1 Purpose and Requirements

The `luanumbers` package is intended for documents whose decimal precision should be controlled centrally. After one preamble setup, it adjusts eligible decimal literals automatically across the document body. Authors can therefore revise the document-wide precision without finding and rewriting each number or marking every number with a special command. Exact decimal-string arithmetic is used rather than binary floating point. The package requires LuaLaTeX.

Important

Automatic processing is intentionally conservative. Structural data, protected commands, protected environments, URLs, versions, and other ambiguous values are preserved. Use `\LuaNumber` when formatting must be explicit.

2 Installation and Compilation

Place `luanumbers.sty` and `luanumbers.lua` beside the document, or install both files in a local TeX tree. Compile with:

```
lualatex document.tex
```

The package cannot be used with pdfLaTeX or XeLaTeX.

3 Basic Setup

LaTeX input

```
\usepackage{luanumbers}

\LuaNumbersSetup{
  decimals=1,
  rounding=half-up,
  pad-zeroes=true,
  integers=false,
  warnings=once
}
```

With this configuration, decimal literals are shown with one decimal place, while integers remain unchanged.

4 Input and Output

LaTeX input

```
Pi is approximately 3.14159265.
The measured value is 3.00 units.
The correction is -1.234 units.
There are 7 complete samples.
```

Processed output

Pi is approximately 3.1. The measured value is 3.0 units. The correction is -1.2 units. There are 7 complete samples.

The decimal values become `3.1`, `3.0`, and `-1.2`. The integer `7` is not changed.

4.1 Mathematics

LaTeX input

```
\[
  1.23456 + 9.87654 = 11.11110
\]
```

Processed output

$$1.2 + 9.9 = 11.1$$

4.2 Scientific Notation

Scientific notation is preserved by default. Thus `6.0e2` becomes `6.0e2` with the one-decimal setup. A literal such as `6e2` is unchanged when `integers=false`.

4.3 Explicit Formatting

`\LuaNumber` always formats its argument using the active settings:

```
The result is \LuaNumber{12.3456}.
```

It is also the preferred command for values in ambiguous contexts or protected environments.

5 Configuration Reference

Setting	Default	Accepted values
`decimals`	`2`	Integer from 0 to 100
`rounding`	`half-up`	`half-up`, `half-even`, `truncate`, `floor`, `ceil`
`pad-zeroes`	`true`	Boolean
`integers`	`false`	Boolean
`preserve-exponent`	`true`	Boolean
`normalize-negative-zero`	`true`	Boolean
`input-decimal`	`dot`	`dot`, `comma`, `both`
`auto-protect`	`true`	Boolean
`warnings`	`once`	`off`, `once`, `all`, `error`

5.1 Rounding Modes

`half-up` rounds a decimal tie away from zero. `half-even` rounds a tie toward an even final digit. `truncate` discards excess digits. `floor` rounds toward negative infinity and `ceil` rounds toward positive infinity.

6 Exact Decimal Arithmetic

The formatter operates on digit strings. It therefore handles values such as `2.7`, very large decimal strings, and halfway cases without first converting them to a Lua floating-point number. It also supports leading decimals such as `0.1`, decimal commas when configured, Unicode minus signs, and negative-zero normalization.

Automatic mode does not interpret a trailing form such as `3.` because it is indistinguishable from sentence punctuation. The explicit command `\LuaNumber{3.}` can format that form safely.

7 Local Object Exclusion

The recommended way to preserve one particular object is the built-in `luanumbersexclude` environment. Only source inside that specific wrapper is excluded. Other sections, figures, tables, and objects of the same type continue to use automatic rounding.

The wrapper can contain the object's heading, caption, label, body, and nested environments. All numeric source within it is copied to TeX unchanged. Generated metadata files such as `.aux`, `.toc`, `.lof`, and `.lot` are never rewritten, so protected headings and labels remain stable across repeated LaTeX passes.

7.1 One Selected Section

```
\begin{luanumbersexclude}
\section{Experimental results 3.14159}
\label{sec:results-3.14159}

This selected section preserves 3.14159 exactly.
\end{luanumbersexclude}

\section{Discussion 3.14159}
This later section is processed normally: 3.14159.
```

The first heading, label, and section body are preserved. The second section is outside the wrapper, so its decimal literals use the configured precision. The exclusion ends exactly at `\end{luanumbersexclude}`; it does not alter all other section commands.

7.2 One Selected Figure

```
\begin{luanumbersexclude}
\begin{figure}
  \centering
  \includegraphics[width=0.75\textwidth]{result-3.14159.pdf}
  \caption{Unmodified measurement 3.14159}
  \label{fig:result-3.14159}
\end{figure}
\end{luanumbersexclude}
```

Only this figure is isolated. Its graphics options, file name, caption, label, and body remain unchanged. A later figure outside the wrapper is unaffected by this exclusion policy.

7.3 One Selected Table

```
\begin{luanumbersexclude}
\begin{table}
  \centering
  \caption{Raw values from dataset 2.7}
  \label{tab:raw-2.7}
  \begin{tabular}{cc}
    1.2345 & 9.8765 \\
  \end{tabular}
\end{table}
\end{luanumbersexclude}
```

This preserves the selected table's data and metadata without disabling automatic rounding in other tables.

Important

The `\label` command is protected by default even without a local wrapper, so label keys such as `fig:result-3.1` are not rewritten. The local wrapper additionally preserves the selected object's visible caption, heading, body, and other numeric source.

8 Globally Protected Environment Types

The package protects these environments by default:

`tikzpicture`, `axis`, `pgfpicture`, `verbatim`, `verbatim*`, `lstlisting`, `minted`, `filecontents`, and `filecontents*`.

All numeric source inside a protected environment is passed through unchanged. For graphics environments, this refers only to the package's *automatic input rewriting*. The explicit `\LuaNumber{...}` command still works inside a protected environment because TeX executes that command after the input line has been preserved. This distinction prevents coordinates and data from being changed while still allowing selected visible labels to be rounded. Register one environment or a comma-separated list in the preamble only when *every instance* of that environment type should be excluded:

```
\LuaNumbersProtectEnvironment{mydiagram}

\LuaNumbersProtectEnvironments{
  mydiagram
}
```

Remove protection with the corresponding singular or plural command:

```
\LuaNumbersUnprotectEnvironment{mydiagram}
\LuaNumbersUnprotectEnvironments{mydiagram,otherdiagram}
```

Important

Registering `figure` or `table` here is global: every matching float in the document is preserved. To isolate one particular figure or table, use `luanumbersexclude` around that object instead.

9 Protected Commands

Arguments of common structural commands are protected by default, including references, citations, URLs, file names, graphics options, length settings, and TikZ/PGF setup commands.

Document-specific commands can be registered in the preamble when every use of that command should have protected arguments:

```
\LuaNumbersProtectCommands{
  caption,
  custommetadata
}
```

This applies globally to every registered command. Use the local `luanumbersexclude` wrapper to preserve only one particular section or caption. Command names may be written with or without their leading backslash. Multiline braced arguments are tracked until their closing brace.

The related singular and removal operations are:

```
\LuaNumbersProtectCommand{command}
\LuaNumbersUnprotectCommand{command}
\LuaNumbersUnprotectCommands{command-a,command-b}
```

10 TikZ, PGFPlots, and Beamer

The package does *not* automatically rewrite numeric literals inside `tikzpicture`, `pgfpicture`, or `axis`. This is deliberate: the same literal may be a coordinate, dimension, transformation, plot datum, or style parameter, and rounding it could alter the graphic rather than merely its printed text.

The resulting behavior is:

Source context	Automatic rounding	Recommended control
Ordinary text outside graphics	Yes	`\LuaNumbersSetup`
TikZ coordinates and dimensions	No	Leave protected
Plain numeric text in a TikZ node	No	Use `\LuaNumber`
PGFPlots coordinates and tables	No	Leave protected
PGFPlots-generated tick labels	No	Use PGFPlots number formatting

Explicit `\LuaNumber` calls are still processed inside protected graphics environments. For example, the geometry below is copied unchanged, but the visible value in the node is rounded to the package precision:

```
\begin{tikzpicture}
  \draw (0.00,0.00) rectangle (3.26,1.74);
  \node at (1.63,0.87)
    {Value: \LuaNumber{3.14159}};
\end{tikzpicture}
```

Thus, with one decimal place, `(3.26,1.74)` remains exactly a TikZ coordinate while the node displays `Value: 3.1`. A plain node such as `\node {Value: 3.14159};` would retain `3.14159`, because it has not requested explicit formatting.

PGFPlots creates tick labels from plot data after reading the source. Configure those labels through PGFPlots rather than by changing the underlying data:

```
\pgfplotsset{
  tick label style={/pgf/number format/.cd,
    fixed, fixed zerofill, precision=1}
}
```

It is possible to opt into automatic source rewriting globally:

```
\LuaNumbersUnprotectEnvironment{tikzpicture}
\LuaNumbersUnprotectEnvironment{axis}
```

This affects every numeric literal in every matching environment, including coordinates and dimensions, so it is not the recommended method for formatting displayed labels.

The same behavior works when TikZ is nested inside a Beamer frame. Ordinary frame text remains subject to automatic rounding, while the graphics source is preserved and explicit `\LuaNumber` calls inside the graphic still work.

11 Warnings and Ambiguous Input

Automatic mode leaves suspicious values unchanged and issues a warning when a number appears in a context resembling a URL, date, ratio, key-value setting, grouped number, or version. Examples include:

```
https://host/3.14159
key=3.14159
1,234.56
1.2.3
```

Set `warnings=error` during validation or continuous integration to stop compilation whenever such a case requires review. Use `\LuaNumber` when the suspicious value is definitely display text.

12 Manual Suspension

Automatic processing can be suspended for exceptional source regions:

```
\LuaNumbersOff
This value is preserved exactly: 3.14159265.
\LuaNumbersOn
```

Place these commands on separate source lines. Registered environment and command protection is preferable for recurring structures, while `luanumbersexclude` is preferable for one selected document object.

13 Limitations

The package processes LuaTeX input lines before TeX expands macros. It cannot infer semantic intent from arbitrary macro-generated source, numbers assembled from several macros, or environments whose begin/end commands are generated indirectly. Automatic processing is therefore a convenience layer, while `\LuaNumber` remains the deterministic interface.

When `integers=true`, integer arguments expected by LaTeX commands may be converted to decimal strings. Use that setting cautiously and protect structural commands that require literal integers.

14 Development and Verification

From the project root:

```
make doc          # build this manual
make examples     # build TikZ/PGFPlots and Beamer examples
make test         # run Lua assertions and compile the smoke test
make clean        # remove generated TeX files
```