The luaplot package in LaTeX

Chetan Shirore and Ajit Kumar

July 20, 2023

1 Introduction

The `luaplot` package is developed using Lua to plot graphs of real-valued functions of a real variable in LaTeX. It is developed with the `MetaPost` system [4] and `luamplib` [3] and `luacode` [2] packages. It provides an easy way for plotting graphs of standard mathematical functions and their finite combinations. There is no particular environment in the package for plotting graphs. It also works inside floating environments of LaTeX like tables and figures. The compilation time to plot several graphs in LaTeX using the luaplot package is significantly less with LuaLaTeX engine.

The package is based on the core idea of loading mathematical functions inside Lua and determining plot points using different methods available in Lua. After determining plot points in Lua, two different approaches are used:

- parse plot points to the MetaPost system via `luampblib`.

- parse plot points to the `tikz` package.

The MetaPost system is based on the Metafont to produce precise technical illustrations. Donald Knuth designed Metafont for TeX. John Hobby designed the MetaPost system to produce scalable PostScript or scalable vector graphics. The output from MetaPost can be directly included with LaTeX. The first approach thus offers a native way of plotting graphs inside LaTeX using Lua and MetaPost.

Tikz is designed by Till Tantau for producing vector graphics from different expressions. Drawing lines, arrows, paths, geometric shapes, etcetera is possible using Tikz [5]. Tikz commands can be considered TeX macros, but Tikz itself is a language. LaTeX users widely use Tikz to produce different graphics. The second approach combines Lua and Tikz to plot graphs inside LaTeX.

2 Installation and License

The installation of `luaplot` package is similar to plain latex package, where the `.sty` file is in LaTeX directory of texmf tree. The package can be included with `\usepackage{luaplot}` command in the preamble of the LaTeX document. A TeX file is to be compiled using the LuaLaTeX engine.

The `luaplot` package is released under the LaTeX Project Public License v1.3c or later. The complete license text is available at http://www.latex-project.org/lppl.txt. It is developed in Lua. Lua is available as a certified open-source software. Its license is simple and liberal, which is compatible with GPL.

3 The luaplot command

The `luaplot` command plots graph of a real valued function of a real variable in LaTeX using Lua and the MetaPost system. The plot points are exclusively determined using Lua. The algorithm is quite efficient to calculate coordinates of points for plotting in the MetaPost. It is deliberately written in Lua. The graphing is done using the MetaPost system. The `luaplot` command has the following syntax. It uses `xkeyval` package [6] for dealing with optional parameters.

```
\luaplot[optional parameters]{function(s)}
```

The only compulsory argument is the function(s). Multiple functions can be given as input separated by a comma. The standard functions available from mathematics library are *abs, acos, asin, atan, ceil, cos, exp, floor, log, sin, sqrt, tan,* etcetera. The custom functions can easily be defined inside the luacode package. Table 1 lists optional parameters in the `luaplot` command with their brief description.

Optional Parameter	Description
xmin	Specifies the minimum value of x coordinate used for calculating plot points. The default value of xmin is 0.
xmax	Specifies the maximum value of x coordinate used for calculating plot points. The default value of xmax is 5.
ymin	Specifies the minimum value of y coordinate used for calculating plot points. If not specified, it is determined by values of xmin and xmax.
ymax	Specifies the maximum value of y coordinate used for calculating plot points. If not specified, it is determined by values of xmin and xmax.
plotpts	Specifies the number of plot points used to plot graphs of functions. The default value of `noofpoints` is 100.
hor	Specifies the width of plotting window. The default value of width is 5cm. It can be specified in any unit that LaTeX accepts: *pt*(point), *pc*(pica), *in*(inch), *cm*(centimeter), *mm*(millimeter), etcetera.
ver	Specifies the height of plotting window. The default value of height is 3cm. It can be specified in any unit that LaTeX accepts: *pt*(point), *pc*(pica),*in*(inch), *cm*(centimeter), *mm*(millimeter), etcetera.
plotsty	Specifies plotting styles of curves. Styles for different curves are to be given in curly braces separated by comma. The basic syntax is `withpen<pen expression> | dashed<picture expression>` It can be scaled also. For all possible styles, the MetaPost package [4] can be referred.
clr	Specifies colors for plotting. Colors for different curves are to be given in curly braces separated by semicolon. The default color is black. Colors can be specified in a variety of ways. Standard colours can be specified by using names such as "red"; "blue"; "green" etcetera. The RGB format (r,g,b) can also be used.
plotoptions	Specifies different options for plotting. The input is to be in curly braces. If the expression is a big string and multi-line then it can be enclosed in two square brackets (string in Lua). The different plot options include grids, labels, legends, axes styles, point markings on axes, framing styles etcetera. For all possible options, the MetaPost package documentation [4] and guide [1] can be referred.

Table 1: Optional parameters in the luaplot command

4 The luatikzpath command

The `luatikzpath` command produces a sequence of points for plotting that can be used inside tikz. The best feature is that all plotting options in tikz remain with the user. The all custom formatting can be done with
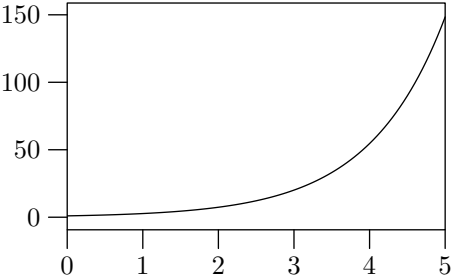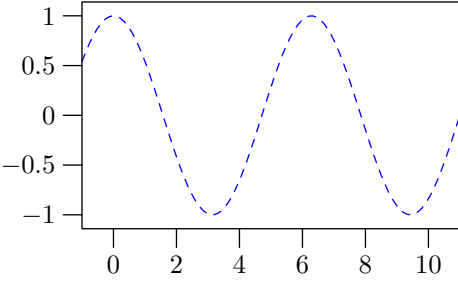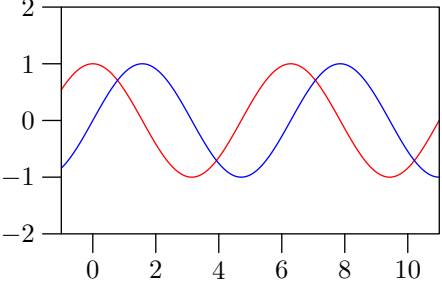
draw options, node options, tikz styles etcetera in tikz. The sequence of points is entirely determined using Lua. The `luatikzpath` command in luaplot package has the following syntax.

`\luaplot{function}{xmin}{xmax}{plot points}`

The first compulsory argument is the function. The standard functions available from mathematics library: *abs, acos, asin, atan, ceil, cos, exp, floor, log, sin, sqrt, tan,* etcetera can be input. The second and third compulsory arguments are `xmin` and `xmax`. These specify the beginning and end values of x coordinate. The last compulsory argument is the number of points used for plotting.

5   Plotting graphs with the luaplot command

Table 2 illustrates plotting graphs of functions with the `luaplot` command.

LaTeX Input	Graph Produced
`\luaplot{exp(x)}`	

`\luaplot[`
`    xmin=-1, xmax=11,`
`    clr={ 'blue' },`
`    plotsty={'dashed evenly'}`
`]{cos(x)}`	

`\luaplot[`
`    xmin=-1, xmax=11,`
`    ymin=-2, ymax=2,`
`  plotpts=1000,`
`    clr={ 'red; blue' }`
`]{cos(x),sin(x)}`	

```
\luaplot[
clr={ 'red; green;
  (0.58,0.2,0.62)' },
xmin=-10,xmax=10,
ymin=-10,ymax=10]
{x^3,-x^3,x}
```
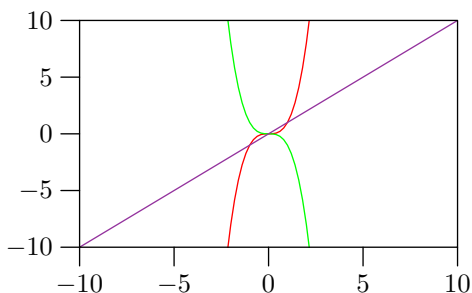
```
\luaplot[xmin=-5,xmax=5,
clr={'red; blue'}]
{(2*(x^2)-1)/(x^2-1),1/x}
```
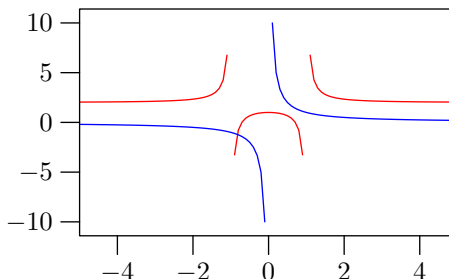
Table 2: Plotting graphs with the luaplot command

Listing 1 illustrates the use of optional argument `plotoptions` in the `luaplot` command. It generates graphs shown in Figure 1.

Listing 1: Plotting with the luaplot command

```
\luaplot[
xmin=-10,
xmax=20,
plotpts=300,
hor='6cm',
ver='5cm',
clr={'red; blue'},
plotsty={'dashed evenly'},
plotoptions={
[[
glabel(btex ${sin(x)+x}$ etex,(7,11.5));
glabel(btex ${cos(x)}$ etex,(12,2));
glabel.bot(btex Some Graphs etex, OUT);
autogrid(grid.bot,) withcolor .85white;
autogrid(grid.lft,) withcolor .85white;
frame.dashed evenly;
]]
}
]
{sin(x)+x,cos(x)}
```
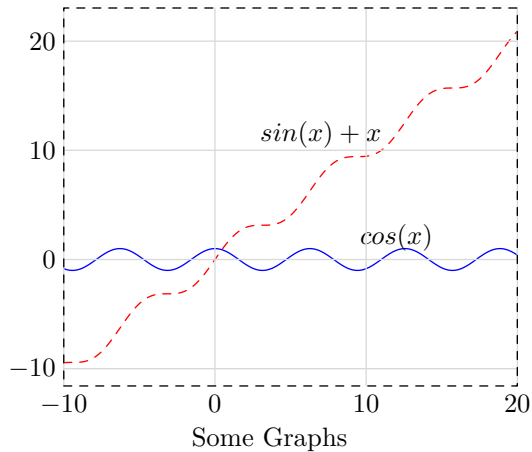
Figure 1: Plotting graphs with the luaplot package

6 Plotting graphs with the luatikzpath command

Listing 2 illustrates plotting graph of a function with the `luatikzpath` command. Multiple graphs can be plotted in a single picture environment.

Listing 2: Plotting with the luatikzpath command

```
\begin{tikzpicture}
\draw[thin,->] (-4.2,0)--(5,0)node[right]{$x$};
\draw[thin,->] (0,-3)--(0,2.5)node[above]{$y$};
\draw[red] \luatikzpath{sin(x^2)}{-4}{4}{100} node at (1,1.3) {$y=\sin(x^2)$};
\draw[blue] \luatikzpath{log(x)}{0.1}{4.5}{100} node at (4,1.7) {$y=\log(x)$};
\end{tikzpicture}
```
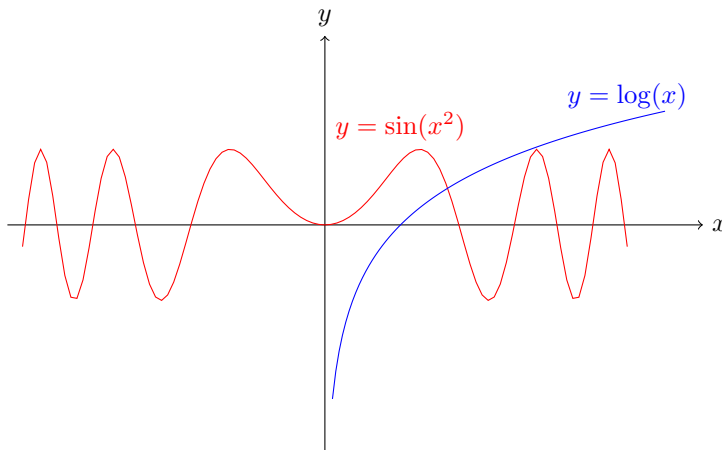
Listing 2 generates graphs shown in Figure 2.

Figure 2: Plotting with the luatikzpath command

5

7 Known issues and limitations

The package does not use any external library supporting arbitrary precision arithmetic. The luaplot package can handle big and small numbers within the range of Lua that it supports. However, the MetaPost system does not support numbers in scientific notation. The coordinates of plot points produced in scientific notation are rounded off to 12 significant decimal places within the package. This may cause slight deviations from actual values. The same issue is not faced while parsing points to tikz as it supports input in scientific notation.

References

[1] John D. Hobby. *Drawing Graphs with MetaPost*. URL: https://tug.org/docs/metapost/mpgraph.pdf.

[2] *luacode package*. visited on 2022-03-10. 2012. URL: https://mirror.kku.ac.th/CTAN/macros/luatex/latex/luacode/luacode.pdf.

[3] *luamplib package*. 2022. URL: https://mirror.kku.ac.th/CTAN/macros/luatex/generic/luamplib/luamplib.pdf (visited on 02/22/2022).

[4] *MetaPost system*. 2020. URL: https://www.tug.org/docs/metapost/mpman.pdf (visited on 02/22/2022).

[5] *pgf Package*. visited on 2021-01-10. URL: https://ctan.org/pkg/pgf?lang=en.

[6] *xkeyval Package*. visited on 2021-03-10. URL: https://ctan.org/pkg/xkeyval.