# Example using Mathematica to generate similar plot as Tikz using Lua

Nasser M. Abbasi

January 31, 2024 Compiled on January 31, 2024 at 12:01am

This note show how to generate Latex document in Mathematica that gives same output as code I saw using Lualatex in paper Numerical metods with Lualatex [1]. The idea of the above paper is to use lua code to assist in making the computation and then use Tikz for the plotting part. Lua makes it easier to do the computation to generate the data to be plotted, then the data is passed back to tikz for plotting.

There are a number of examples given in the above paper. Below I show how to generate complete Latex document from inside Mathematica which gives the same plot for one example. Later I will add the other examples.

The advtange of using Mathematica to generate the complete Latex document, is that the computation is done in the same notebook, and Mathematica is much more powerfull at computation. I have used this method before to generate large Latex reports, all from inside Mathematica.

Ofcourse, one have to generate the Latex code using Mathematica strings. But this is not a problem I found. One has to just escape each Latex "\".

Here is the complete Latex code, taken from the above paper for one example, where I just added the preamble to make it compile as a standalone Latex file. I used TexLive 2016.

```
\documentclass[12pt]{article}%

\usepackage{pgfplots}
\usepackage{tikz}
\usepackage{luacode}
\begin{luacode*}

-- Fourier series
function partial_sum(n,x)
```

```
partial = 0;
for k = 1, n, 1 do
partial = partial + math.sin(k*x)/k
end;
return partial
end
-- Code to write PGFplots data as coordinates
function print_partial_sum(n,xMin,xMax,npoints,option)
local delta = (xMax-xMin)/(npoints-1)
local x = xMin
if option~=[[]] then
tex.sprint("\\addplot["..option..."] coordinates{")
else
tex.sprint("\\addplot coordinates{")
end
for i=1, npoints do
y = partial_sum(n,x)
tex.sprint("("..x..","..y..")")
x = x+delta
end

tex.sprint("}")
end
\end{luacode*}

\newcommand\addLUADEDplot[5][]{%
\directlua{print_partial_sum(#2,#3,#4,#5,[[#1]])}%
}

\begin{document}

\begin{figure}
\pgfplotsset{width=15cm, height=7cm}
\begin{tikzpicture}\small
\begin{axis}[xmin=-0.2, xmax=31.6, ymin=-1.85, ymax=1.85,
xtick={0,5,10,15,20,25,30},
ytick={-1.5,-1.0,-0.5,0.5,1.0,1.5},
minor x tick num=4,
minor y tick num=4,
axis lines=middle,
axis line style={-}
```

```
]
% SYNTAX: Partial sum 30, from x = 0 to 10*pi, sampled in 1000 points
\addLUADEDplot[color=blue,smooth]{30}{0}{10*math.pi}{1000};
\end{axis}
\end{tikzpicture}
\caption{The partial sum $\sum_{k=1}^{30} \frac{\sin(kx)}{k}$ of the Fourier series of
$f(x)=(\pi-x)/2$ illustrating the Gibbs phenomenon.}
\end{figure}

\end{document}
```

Compiling the above using `lualatex tikz.tex`, here is the pdf generated

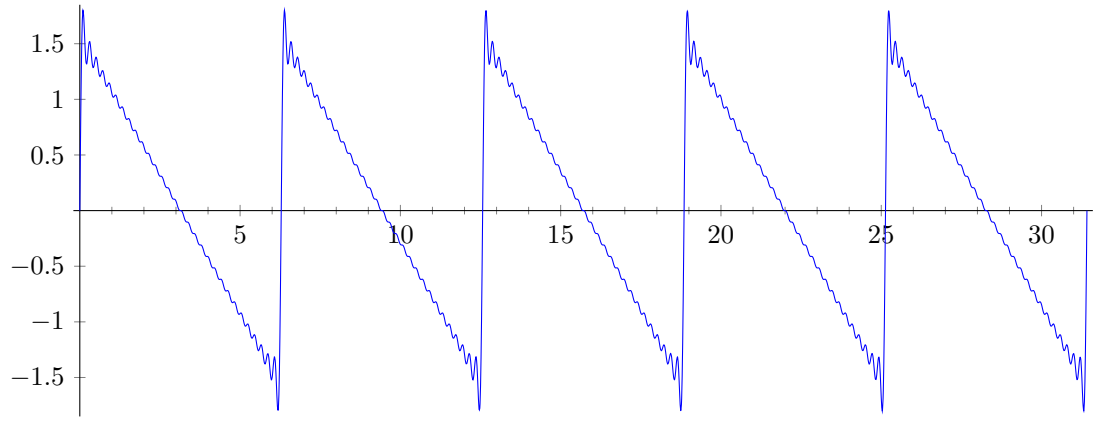


Figure 1: The partial sum $\sum_{k=1}^{30} \frac{\sin(kx)}{k}$ of the Fourier series of $f(x) = (\pi - x)/2$ illustrating the Gibbs phenomenon.

Figure 1: output of tikz.tex, example code from the paper

Next I opened a Mathematica notebook, and wrote the code which generated a full latex file, self contained, which does the same computation and generate the plot. Here is the Mathemmatica code. After Making the plot, the image is saved to pdf file, and then used in the next statement.

```
partialSum[x_,n_]:=Sum[Sin[k x]/k,{k,1,n}];
xMin=-0.2;xMax=31.6;
nPoints=30;nSamples=1000;
delta=(xMax-xMin)/(nSamples-1);
x=xMin;
data=Last@Reap@Do[
```

```
Sow@{x,partialSum[x,nPoints]};
x=x+delta,
{nSamples}
];
g=ListLinePlot[data[[1]],PlotTheme->"Classic",BaseStyle->15,PlotStyle->Blue];
SetDirectory[NotebookDirectory[]];
Export["g.pdf",g];
str="\\documentclass[11pt]{scrartcl}%
\\IfFileExists{luatex85.sty}{\\usepackage{luatex85}}{}

\\ifdefined\\HCode% detect tex4ht
\\usepackage[utf8]{luainputenc}
\\usepackage[T1]{fontenc}
\\else
\\usepackage{fontspec}
\\fi
\\usepackage{graphicx}

\\begin{document}
\\begin{figure}
\\includegraphics[width=0.7\\textwidth]{g}
\\caption{The partial sum $\\sum_{k=1}^{30} \\frac{\\sin(kx)}{k}$ of the Fourier series of
$f(x)=(\\pi-x)/2$ illustrating the Gibbs phenomenon.}
\\end{figure}
\\end{document}";
fileName="mma.tex";
If[FileExistsQ[fileName],DeleteFile[fileName]];
file=OpenWrite[fileName,PageWidth->Infinity];
WriteString[file,str];
Close[file];
```

running the above Mathematica code, it generated the following Latex file called `mma.tex`

```
\documentclass[11pt]{article}%
\usepackage{graphicx}

\begin{document}
\begin{figure}
\includegraphics[width=0.7\textwidth]{g}
\caption{The partial sum $\sum_{k=1}^{30} \frac{\sin(kx)}{k}$ of the Fourier series of
$f(x)=(\pi-x)/2$ illustrating the Gibbs phenomenon.}
\end{figure}
\end{document}
```

Now compiling the above Latex file, also using `lualatex mma.tex` gave this pdf file

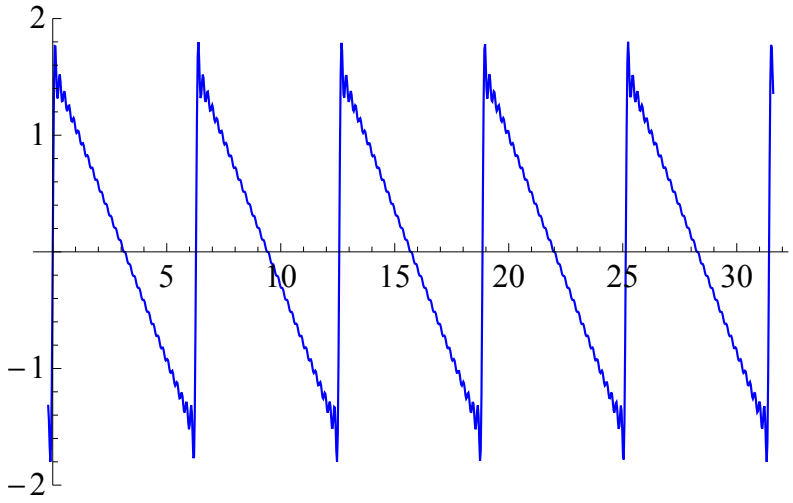


Figure 1: The partial sum $\sum_{k=1}^{30} \frac{\sin(kx)}{k}$ of the Fourier series of $f(x) = (\pi - x)/2$ illustrating the Gibbs phenomenon.

Figure 2: output of mma.tex, Mathematica generated Latex

The main objective to this method might be is that the graphics and the annotations generated by Mathematica is not using native Latex notations and so it might not fit well in the document as those generated by Tikz and Lua code. But this also has a solution, thanks to a new package called Matex.

I did not Matex in this example as it was not needed, but could have easily done so if annotations are needed, and will do that when I add more examples later on for

illustrations. Matex allows one to add Latex annotations to a plot and graphics.

## Reference

This is the paper that I took the lua code above to write equivalent one in Mathematica. A well written and useful paper with many good Lua code examples.

The PracTEX Journal, 2013, No. 1
Article revision 2013/1/27

# Numerical methods with LuaLaTeX

Juan I. Montijano, Mario Pérez, Luis Rández and Juan Luis Varona

Email {monti,mperez,randez}@unizar.es and jvarona@unirioja.es
Address Universidad de Zaragoza (Zaragoza, Spain) and Universidad de La Rioja (Logroño, Spain)

Abstract An extension of TeX known as LuaTeX has been in development for the past few years. Its goal is to allow TeX to execute scripts written in the general purpose programming language called Lua. There is also LuaLaTeX, which is the corresponding extension for LaTeX.

In this paper, we show how LuaLaTeX can be used to perform tasks that require a large amount of mathematical computation. With LuaLaTeX instead of LaTeX, we achieve important improvements: since Lua is a general purpose language, rendering documents that include evaluation of mathematical algorithms is much easier, and generating the pdf file becomes much faster.