LuaTeX

Using LuaTeX for elegant scientific typesetting

SEPTEMBRSE

THE VIRTUAL RSE CONFERENCE

Legacy

- LaTeX widely used in academia
- Frustration: Hard to modify the default layout
- Tricky macro definition
- Search for packages
- Seemingly ad-hoc solutions

Solution

- LuaTeX = typesetting system + Lua scripting engine
- Write cleaner code
- Dynamic contents
  - calculation
  - geometry

Output

PDF

- Universal format
- Font handling
- Compressed
- Security-enabled
- Extendable to include multimedia

luacode package

```
\luadirect
\luaexec
\luastring
luacode
luacode*
tex.sprint
tex.write
```

Text

Print TeX

```
\begin{luacode}
 tex.sprint("\string\\pi
    \string\\neq",
    tostring(math.pi))
\end{luacode}
```

$\pi \neq 3.14159265358898$

Macros allowed

```
\newcommand\two{2}
\begin{luacode}
    tex.sprint("The
square root of two is:
", math.sqrt(\two))
\end{luacode}
```

The square root of two is: 1.4142135623731

Randomize for making quizzes

Tables

Reading data from files

- File utility
- **for** loops
- String manipulations

Example: Near-earth comets

```
\begin{luacode*}
function readfile(filename, n, m)
  ...
end
\end{luacode*}
```

```
% macro
\newcommand{\readfile}[3]{\luadirec
t{readfile(#1,#2,#3)}}
% TeX code
\begin{tabular}{*{10}{c}}
  \readfile{"near-earth-
comets.csv"}{5}{3}
\end{tabular}
```

Object	Epoch	TP
1P/Halley	49400	2446467.395
2P/Encke	56870	2456618.204
3D/Biela	-9480	2390514.115
5D/Brorsen	7440	2407439.534

Total: 5 lines.

Generating number tables

- Functions
- **for** loops

Example: Trigonometry

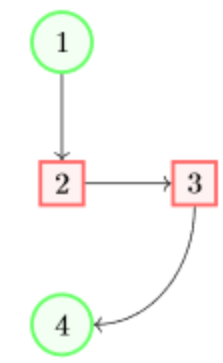
```
function trigtable()
  for t = 0, 45, 3 do
    x = math.rad(t)
    tex.sprint(string.format(
     '%2d° & %1.9f \\\\',
      t, x, math.sin(x)))
  end
end
```

	x	$\sin(x)$
0°	0.000000000	0.000000000
3°	0.052359878	0.052335956
6°	0.104719755	0.104528463

Graphics

Use loops and draw commands

- mplib
- TikZ and PGFplot



Runge-Kutta solver

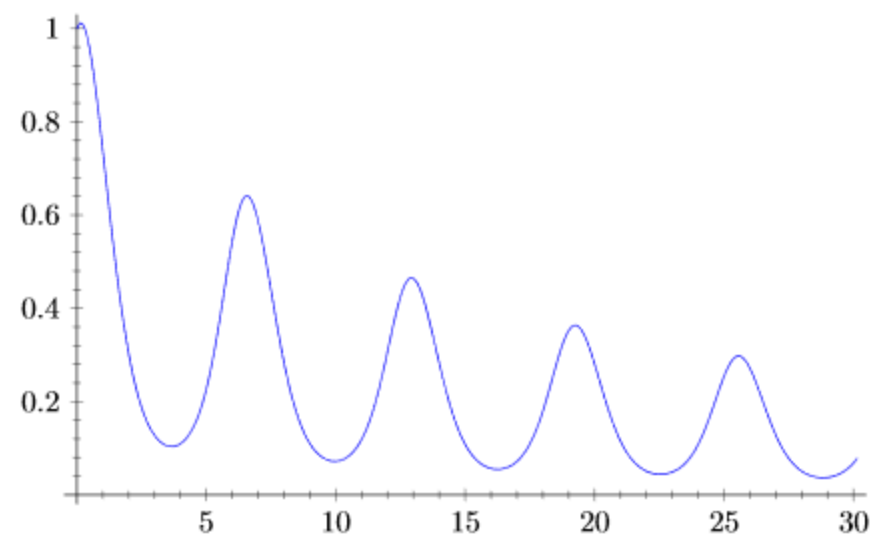
- Define function
- Plot

```
\begin{luacode*}
-- Define math function
function f(t,y)
  return y * math.cos(t+math.sqrt(1+y))
End

-- Solving the ODE equation for the plotting
coordinates
function print_RKfour(tMax,npoints,option)
  t = ...
  y = ...
  tex.sprint("(..t..","..y..")")
end
\end{luacode*}

% the macro part
\newcommand\addLUADEDplot[3][]{\directlua{pri
nt_RKfour(#2,#3,[[#1]])}}

% the plotting part
\begin{tikzpicture} ...
\addLUADEDplot[color=blue,smooth]{30}{200}
...
```



References

- Lamport (1994) LaTeX User's guide and reference manual.
- Ierasulimschy (2016) Programming in Lua
- Pégourié-Gonnard (2012) luacode package
- Montijano et al (2013) PracTeX journal
- Isambert (2010) TUGBoat 31(3)
- Isambert (2011) TUGBoat 32(1)
- Crémer (2011) A very minimal introduction to TikZ
- Overleaf TikZ example
- Mark Wibrow's Brillouin Function in TikZ

https://users.aber.ac.uk/ngc2/luatex.pdf

ngc2@aber.ac.uk

+44 0788 342 6572

1872 PRIFYSGOL ABERYSTWYTH UNIVERSITY