Basic Mathematical Computations inside LaTeX using Lua

Chetan Shirore
mathsbeauty@gmail.com
Department of Mathematics
Institute of Chemical Technology
Mumbai, India

Ajit Kumar
a.kumar@ictmumbai.edu.in
Department of Mathematics
Institute of Chemical Technology
Mumbai, India

Abstract

LaTeX is one of the most widely used typesetting systems for scientific publications. Lua programming language is a scripting language which can be embedded across platforms. We have developed Lua-based packages to perform basic mathematical computations inside LaTeX documents. The main aim of this paper is to introduce Lua-based LaTeX packages developed by us. The **luamaths**, **luacomplex**, **luatruthtable**, **luaset** *and* **luagcd** *packages are introduced in this paper with their applications in LaTeX. These packages are designed from a pedagogical perspective. We have built time-saving macros that can assist in the creation of teaching notes.*

1 Background and Introduction

There is a good scope to perform basic mathematical computations in LaTeX using Lua. This approach is suitable for performing standard operations on real and complex numbers, sets, integers, vectors, and matrices. It involves some complex intermingling of Lua and TeX. No particular environment in LaTeX is required to use the developed packages. The core part of programming is done in Lua, and the tex file is to be compiled with the LuaLatex engine. The time required for performing operations in LaTeX documents is not an issue while compiling with LuaLaTeX. There is no need to install Lua on the users' systems as TeX distributions (TeXLive or MikTeX) come bundled with LuaLaTeX.

With LuaTEX [9] and Luacode [3] packages, it is possible to use Lua in LaTeX. TeX or LaTeX has scope for programming [12]. However, with the weird internals of TeX, there are several limitations, especially for performing calculations on numbers in LaTeX documents. The `luacode` [3] package is used in development of packages apart from the `xkeyval` [15] package. Various packages and commands introduced in the paper can further be extended or modified by writing custom Lua programs.

This paper is organized into different sections. The second section summarises the key features of developed packages. The third section discusses the advantages of techniques and methods used in the development of packages. The fourth section describes some limitations of the approach used in the development of packages. The fifth section discusses the prospects of the research work. Applications and usage of developed packages are given in the sixth section.

2 Lua-based packages in LaTeX

This section provides brief information on **`luamaths`**, **`luacomplex`**, **`luatruthtable`**, **`luaset`**, and **`luagcd`** packages, along with their key features.

2.1 The luamaths package

The **`luamaths`** package [6] provides commands to evaluate trigonometric, inverse trigonometric, logarithmic, and exponential functions. The random numbers can be generated by using commands in the package. There are functions in the package to find the maximum and minimum of finite subsets of real numbers. The greatest and nearest integer to a real number can be determined, and rounding off real numbers is also possible. Numbers can also be entered in scientific notation. For example, the number 1230000 can be entered as $1.23e6$, and the number 0.000123 can be entered as $1.23e-4$. The package provides the standard constant π and the Euler number e. These constants can be used with other commands of the package. The package can also deal with ∞. Table 1 in subsection 6.1 illustrates some commands in **`luamaths`** package.

2.2 The luacomplex package

The **`luacomplex`** package [4] is developed to define complex numbers and perform basic arithmetic operations on complex numbers in LaTeX. It is built over the **`luamaths`** package. The package provides commands to find modulus, principal arguments, complex conjugates, and real and imaginary parts of complex numbers. The package can be modified or extended by adding custom Lua programs. Subsection 6.2 illustrates some commands of the **`luacomplex`** package.

There are a few limitations of the **`luacomplex`** package. Complex numbers are to be defined explicitly by listing real and imaginary parts. It does not support symbolic calculations on complex numbers. These issues can be addressed in future package updates.

2.3 The luatruthtable package

The **`luatruthtable`** package [10] is designed to generate truth tables of logical operators. The package supports nine operations: *not, and, nand, or, xor, implies, iff, nor, xnor*. There is no in-built function in Lua to convert decimal numbers to binary numbers. The function **`toBinary(m,n)`** is constructed in Lua and used to produce a sequence of *True* and *False* values of boolean variables. This function converts a decimal number m to a binary number, say b. If the number of digits in b is less than n, then (n - no. of digits in b) leading zeroes are added to b. Each digit is stored in a table in Lua. This function runs inside a loop where m takes values from 1 to 2^n; n is the number of truth variables. It produces 2^n combinations of n truth variables. Subsection 6.4 illustrates some commands of the **`luatruthtable`** package.

The associativity and precedence of operators still need to be supported. The package can produce correct results only when parentheses are used for each operation. This point is of utmost importance in using the package. There is no native way of defining custom operations in Lua [2]. However, some metamethods can be nested in a way to replicate the operators. All operators defined in this package are instances of such nestings. The question may be raised about a better way of accomplishing this in Lua. There are alternative ways of doing it. For example, instead of defining **logand** operator and using it in the fashion *p*logand*q*, one could define the function *logand* that takes two arguments and use it in a way *logand(p, q)*. However, when it comes to embedding it in LaTeX, one has to use more and more nested parentheses as the number of statements and operations increase. This is the exact reason why such an approach is not used in the package. It sounds more natural to use *(p *logand* (q*logor*r))*implies* s* instead of using *implies(logand(p, logor(q, r)), s)*.

2.4 The luaset package

The **`luaset`** package [8] is developed to define finite sets and perform operations on them in LaTeX. There is no particular environment in the package for performing set-operations. The package supports the nesting of commands for multiple set operations. Sets need to be defined explicitly. There are commands to determine the cardinal number of sets and check whether two sets are equal. It is also possible to check whether the specified element is in a set. Subsection 6.3 illustrates some commands of the **`luaset`** package.

2.5 The luagcd package

The **`luagcd`** package [5] is designed to find the **greatest common divisor (gcd)** of integers in LaTeX. The package also provides a command to obtain the step-by-step computation of gcd of two integers (Table 4) by the Euclidean algorithm [13]. In addition, the package also has the command to express the gcd of two integers as an integer linear combination (Table 5) and thus verify Bezout's identity. It also provides a step-by-step solution to find integers a and b such that $am+bn=d$, where d is gcd of m and n (Table 6). Subsection 6.5 illustrates some commands of the **`luagcd`** package.

3 Techniques and Methods Used

- The techniques and methods used in developing packages do not need any special resources. All the techniques are platform-independent and work on most standard operating systems, including Microsoft Windows, Linux and its derived versions, and Macintosh. The technique mainly uses Lua and LaTeX, which are platform-independent and do not need strong hardware resources.
- No proprietary software and tools are used in the development of the packages. All tools are freeware and open source. The developed packages are made available as freeware and open source. This is important as the research work can further be extended without any restrictions. This is within the philosophy of open source and freeware tools that are available for the mathematics community.

4 Known Issues and Limitations

The error-handling mechanism in developed packages is not robust. Some error-handling mechanisms for input values are available in packages. If something absurd is entered in a command, it converts it into *nil* through the *table.pack* function in Lua and processes the rest of the input, or it throws an appropriate error through functions in the package or through inbuilt functions of Lua. The error-handling mechanism of the package can be strengthened in future updates.

The number of operations in each of the packages is limited. Also, symbolic calculations still need to be supported. More operations and symbolic calculations can be supported in future updates.

5 Future Plans and Prospects of the Research Work

The approach of embedding Lua in LaTeX can be extended in several other ways. With `luamplib` [7] and `metapost` [11] libraries, it is possible to plot graphs of functions in LaTeX in a native way. This sort of package can be developed in the future to extend the series of Lua-based packages in LaTeX.

There is wide use of linear algebra in several fields and often a need to type vectors and matrices inside LaTeX documents and perform different operations on them. The linear algebra specific packages of LaTeX can be developed using Lua. These packages will be developed in the next part of the research work.

The powerful programming capabilities of Lua can also be used to produce customized documents from LaTeX files. This approach can further be extended to produce Web Apps and Android Apps from LaTeX files. The mathematical content in LaTeX files can be pre-processed and pre-compiled in Web and Android Apps using Lua. We will develop Lua-based techniques to create custom documents and produce Web and Android Apps from LaTeX files. This approach is very innovative and can have several benefits over other techniques of creating Web and Android Apps for mathematical content. One of the android applications developed by us based on this technique is available on the Google play store: "Sequences of Real Numbers" [14]. The technique can be refined to automate the creation of Android applications from LaTeX files. The sample web application produced from LaTeX file can be found on this link [1].

6 Applications and Usage of Developed Packages

The various commands in **`luamaths`**, **`luacomplex`**, **`luaset`**, **`luatruthtable`**, and **`luagcd`** packages are illustrated in subsections of this section.

6.1 Illustrations of commands in the luamaths package

Table 1 illustrates various operations in the **`luamaths`** package.

LaTeX input	Output
\mathOp{(3+3)^4-4}	1292.0
\mathAbs{-6.96}	6.96
\mathFloor{-6.9}	-7
\mathFloor{9.9}	9
\mathCeil{-9.9}	-9
\mathCeil{6.3}	7
\mathRound{9.678884}{3}	9.679
\mathMax{20, 30, 100}	100
\mathMin{20, 30, 100}	20
\mathSqrt{9}	3.0
\mathRound{\mathPi}{6}	3.141593
\mathInf	Inf
\mathOp{\mathInf/2}	Inf
\mathPi	3.1415926535898
\mathLog{2}	0.69314718055995
\mathExp{3}	20.085536923188
\mathRad{1800}	31.415926535898
\mathSin{0}	0.0
\mathCos{0}	1.0
\mathTan{0}	0.0
\mathAsin{0}	0.0
\mathAcos{0}	1.5707963267949
\mathAtan{0}	0.0
\mathOp{9+\mathCos{0}}	10.0

Table 1: Illustrations of commands in the luamaths package

The package can be extended to define new functions. For example, the code in listing 1 output to 0.5.

Listing 1: Extending luamaths package

```
\documentclass{article}
\usepackage{luamaths}
\begin{luacode}
function avgMinMax(...)
avg = (math.min(...) + math.max(...))/2
return avg
end
\end{luacode}
\newcommand\mathAvgMinMax[1]{\directlua{tex.sprint(avgMinMax(#1))}}
\begin{document}
\mathAvgMinMax{\mathCos{0},\mathSin{0}}
\end{document}
```

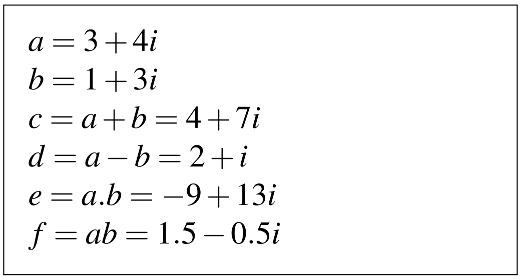
6.2 Illustrations of commands in the the luacomplex package

The latex document (Listing 2) uses various commands from the **luacomplex** package.

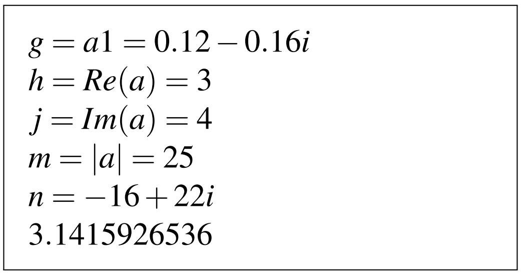
Listing 2: LaTeX document with luacomplex package

```
\documentclass{article}
\usepackage{luacomplex}
\begin{document}
\cpxNew{a}{3, 4}
\cpxNew{b}{1, 3}
\(a=\cpxPrint{a}\) \\
\(b=\cpxPrint{b}\) \\
\cpxAdd{c}{a}{b}
\(c=a+b=\cpxPrint{c}\) \\
\cpxSub{d}{a}{b}
\(d=a-b=\cpxPrint{d}\) \\
\cpxMul{e}{a}{b}
\(e=a. b=\cpxPrint{e}\) \\
\cpxDiv{f}{a}{b}
\(f=\frac{a}{b}=\cpxPrint{f}\) \\
\cpxInv{g}{a}
\(g=\frac{1}{a}=\cpxPrint{g}\) \\
\cpxRe{h}{a}
\(h=Re(a)=\cpxPrint{h}\) \\
\cpxIm{j}{a}
\(j=Im(a)=\cpxPrint{j}\) \\
\cpxMod{m}{a}
\(m=|a|=\cpxPrint{m}\) \\
\cpxOp{n}{a+b*c-d}
\(n=\cpxPrint{n}\) \\
\luaexec{tex. sprint(string. format('\%. 10f', math. pi))}
\end{document}
```

The latex document (Listing 2) outputs the following on compiling with LuaLaTeX engine.



$a = 3+4i$
$b = 1+3i$
$c = a+b = 4+7i$
$d = a-b = 2+i$
$e = a.b = -9+13i$
$f = ab = 1.5-0.5i$



$g = a1 = 0.12-0.16i$
$h = Re(a) = 3$
$j = Im(a) = 4$
$m = |a| = 25$
$n = -16+22i$
3.1415926536

6.3 Illustrations of commands in the the luaset package

The latex document (Listing 3) uses various commands from the `luaset` package.

Listing 3: LaTeX document with luaset package

```
\documentclass{article}
\usepackage{luaset,parskip}
\begin{document}
\luaSetNew{A}{a,b,c,\mathCos{0},d,10,a,30}
\luaSetNew{B}{d,e,f,10,20}
\luaSetUnion{C}{A}{B}
\luaSetIntersection{D}{A}{B}
\luaSetDifference{E}{A}{B}
\luaSetDifference{F}{B}{A}
\(A = \luaSetPrint{A}\) % printing set A
\(B = \luaSetPrint{B}\) % printing set B
\(C = A \cup B = \luaSetPrint{C}\) % printing set C
\(D = A\cap B = \luaSetPrint{D}\) % printing set D
\(E = A-B = \luaSetPrint{E}\) % printing set E
\(F = B-A = \luaSetPrint{F}\) % printing set F

The Cardinality of set \(A\) is \luaSetCardinal{A}.
It is \luaSetSubseteq{A}{B} that \(A \subseteq B \).

It is \luaSetSubseteq{A}{C} that \(A \subseteq C \).

It is \luaSetSubset{A}{B} that \(A \subset B \).

It is \luaSetSubset{A}{C} that \(A \subset C \).

\setIntersection{G}{C}{A}
\(G = C \cap A = \luaSetPrint{G}\) % printing set G

It is \luaSetEqual{A}{B} that \(A=B \).

It is \luaSetEqual{G}{A} that \(G=A \).

It is \luaSetBelongsto{z}{A} that \(z \in A \).

It is \luaSetBelongsto{10}{A} that \(10 \in A \).

\luaSetNew{S}{a, 10, b, x, y, z, 100, 200}
\(S = \luaSetPrint{S}\) % printing set S

\luaSetIntersection{H}{C}{S}
\luaSetUnion{I}{D}{S}
\(H = C \cap S = \luaSetPrint{H}\) % printing set H

\(I = D \cup S = \luaSetPrint{I}\) % printing set I
\end{document}
```

The latex document (Listing 3) output the following.

$A = \{10, 30, 1.0, a, b, c, d\}$
$B = \{10, 20, d, e, f\}$
$C = A \cup B = \{10, 20, 30, 1.0, a, b, c, d, e, f\}$
$D = A \cap B = \{10, d\}$
$E = A - B = \{30, 1.0, a, b, c\}$
$F = B - A = \{20, e, f\}$
The Cardinality of set A is 7.
It is false that $A \subseteq B$.
It is true that $A \subseteq C$.

It is false that $A \subset B$.
It is true that $A \subset C$.
$G = C \cap A = \{10, 30, 1.0, a, b, c, d\}$
It is false that $A = B$.
It is true that $G = A$.
It is false that $z \in A$.
It is true that $10 \in A$.
$S = \{10, 100, 200, a, b, x, y, z\}$
$H = C \cap S = \{10, a, b\}$
$I = D \cup S = \{10, 100, 200, a, b, d, x, y, z\}$

6.4 Illustrations of commands in the luatruthtable package

The *luatruthtable* package can accept a finite number of variables. It supports a finite number of variables that one would need.

Listing 4: LaTeX document with luatruthtable package

```
\begin{tabular}{|cc|c|c|c|c|c|c|c|c|c|}
\hline
\(p\) & \(q\) & \(\neg p\) & \(p \land q\) & \(p\lor q\) & \(p \rightarrow q\)
 & \(p \leftrightarrow q\) & \( p\) nand \(q\) & \(p\) xor \(q\) & \( p\) nor
    \(q\) & \(p\) xnor \(q\) \\
\hline
\luaTruthTable{p, q}{lognot*p, p*logand*q, p*logor*q, p*imp*q, p*iff*q,
p*lognand*q, p*logxor*q, p*lognor*q, p*logxnor*q } \\
\hline
\end{tabular}
```

The LaTeX code (Listing 4) generates the output shown in Table 2.

p	q	$\neg p$	$p \land q$	$p \lor q$	$p \rightarrow q$	$p \leftrightarrow q$	p nand q	p xor q	p nor q	p xnor q
F	T	T	F	T	T	F	T	T	F	F
T	F	F	F	T	F	F	T	T	F	F
T	T	F	T	T	T	T	F	F	F	T
F	F	T	F	F	T	T	T	F	T	T

Table 2: The luatruthtable command

6.5 Illustrations of commands in the luagcd package

The command `\luagcd` is illustrated in Table 3.

LaTeX input	Output
`\luagcd{20,30,60,70}`	10
`\luagcd{0,10,75,90}`	5
`\luagcd{-30,45,-90}`	15

Table 3: The luagcd command

The command `\luagcdwithsteps{-20}{-6008}` generates the output shown in Table 4.

Step 1: Apply the division algorithm to 6008 and 20.
$6008 = 20(300) + 8$
Step 2: Apply the division algorithm to 20 and 8.
$20 = 8(2) + 4$
Step 3: Apply the division algorithm to 8 and 4.
$8 = 4(2) + 0$
The gcd of -20 and -6008 is the last non-zero remainder and it is 4.

Table 4: The luagcdwithsteps command

The command `\luagcdlincomb{10011}{210}` generates the output shown in Table 5.

The gcd of 10011 and 210 is 3 and the equation $-10011x + 210y = 3$ and has a solution $(x, y) = (3, -143)$.

Table 5: The luagcdlincomb command

The command `\luagcdlincombwithsteps{-10011}{210}` generates the output shown in Table 6.

Step 1:10011 is written as a linear combination of 10011 and 210.
$10011 = (1)(10011) + (0)(210)$
Step 2:210 is written as a linear combination of 10011 and 210.
$210 = (0)(10011) + (1)(210)$
Step 3: The equation in Step 2 is multiplied by 47 and subtracted from the equation in Step 1.
$141 = (1)(10011) + (-47)(210)$
Step 4: The equation in Step 3 is multiplied by 1 and subtracted from the equation in Step 2.
$69 = (-1)(10011) + (48)(210)$
Step 5: The equation in Step 4 is multiplied by 2 and subtracted from the equation in Step 3.
$3 = (3)(10011) + (-143)(210)$
The gcd of -10011 and 210 is 3 and the equation $-10011x + 210y = 3$ and has a solution $(x, y) = (-3, -143)$.

Table 6: The luagcdlincombwithsteps command

7 Acknowledgements

Authors wish to thank the editors for their valuable feedback. We also are immensely grateful to the Institute of Chemical Technology, Mumbai, and K.T.H.M. College, Nashik, for their constant support. These institutes provided excellent infrastructure and unique facilities to conduct our research and supported us in several other ways.

References

[1] *Android applications from LaTeX files using Lua* (visited on 2022-02-22). https://play.google.com/store/apps/details?id=com.unimaths.realseq

[2] *luacustomoperator* (visited on 2022-02-22). http://lua-users.org/wiki/CustomOperators

[3] *Luacode package page* (visited on 2022-03-10). Comprehensive TeX Archive Network. https://ctan.org/pkg/luacode

[4] *luacomplex package page* (visited on 2022-12-29). https://ctan.org/pkg/luacomplex

[5] *luagcd package page* (visited on 2022-12-30). https://ctan.org/pkg/luatruthtable

[6] *luamaths package page* (visited on 2022-12-27). https://ctan.org/pkg/luamaths

[7] *Luamplib package page* (visited on 2022-12-22). https://mirror.kku.ac.th/CTAN/macros/luatex/generic/luamplib/luamplib.pdf

[8] *luaset package page* (visited on 2022-12-28). https://ctan.org/pkg/luaset

[9] *LuaTeX package page* (visited on 2022-01-22). Comprehensive TeX Archive Network. https://ctan.org/pkg/luatex

[10] *luatruthtable package page* (visited on 2022-09-18). https://ctan.org/pkg/luatruthtable

[11] *The Metapost User's Manual* (2020-12-31). https://www.tug.org/docs/metapost/mpman.pdf

[12] William M. Richter, *TEX and scripting languages* (2007-07-13). The TUGboat. https://tug.org/TUGboat/tb25-1/richter.pdf

[13] *The Euclidean Algorithm* (visited on 2021-06-01). https://en.wikipedia.org/wiki/Euclidean_algorithm

[14] *Web applications from LaTeX files using Lua* (visited on 2022-05-03). http://www.kthmcollege.ac.in/Data/Sequences Problems Web App/seqpr.html

[15] *Xkeyval package page* (visited on 2021-06-12). https://ctan.org/pkg/xkeyval