

1 \LaTeX évolue vers Lua \LaTeX

Au commencement, \LaTeX fournissait un fichier dans un format peu commun <fichier> .dvi. Des programmes successifs comme PostScript ont permis de convertir ces fichiers en .ps puis en .pdf devenu rapidement un standard multiplateforme. Depuis ses origines, le succès de \LaTeX est largement dû à la facilité avec laquelle il est possible d'écrire des formules mathématiques, bien avant les extensions fournies par les traitements de texte les plus connus. Cependant, la palette des polices restait limitée, l'insertion de fichiers image nécessitait souvent de les convertir en .ps, et si \LaTeX permettait d'écrire les mathématiques, il ne savait cependant pas bien calculer, juste quelques opérations arithmétiques pouvaient être introduites dans le code du document.

Le format .pdf s'imposant, le moteur pdf \LaTeX est créé fournissant directement après compilation un document <fichier> .pdf. De nombreux packages viennent augmenter les possibilités du programme, la plupart étant compatibles autant avec le moteur \LaTeX d'origine qu'avec pdf \LaTeX . Cependant quelques difficultés demeurent : pdf \LaTeX n'accepte pas les images avec l'extension .ps par exemple, il n'accepte pas non plus le remarquable package de dessin — et donc de construction de graphe de fonction mathématique — PSTricks. Un autre package de dessin est créé : TikZ, incompatible avec le \LaTeX originel mais fonctionnant avec pdf \LaTeX .

Le début du XXI^{ème} siècle marque un tournant pour \LaTeX : le développement de moteurs — à ce jour non encore stabilisés — qui permettent d'augmenter et de faciliter les possibilités de ce formateur de texte :

D'abord avec X \LaTeX qui permet d'utiliser toutes les polices installées sur le système d'exploitation, la plupart des formats image, les packages TikZ et PSTricks pour les dessins. Les polices spécifiques pour les mathématiques sont actuellement en train de se développer grâce au package unicode-math.

Enfin avec Lua \LaTeX très récent qui intègre notamment la possibilité de calculer grâce au langage Lua intégré. Lua \LaTeX n'est pas stabilisé, en plein développement et le programme va encore évoluer : un source créé aujourd'hui peut ne plus fonctionner demain car des commandes auront été modifiées.

Attention : Pour le moment Lua \LaTeX ne fonctionne pas avec PSTricks contrairement à X \LaTeX . Les développeurs du programme Lua \LaTeX devraient à terme trouver une solution à son intégration. Finalement, la solution X \LaTeX apparaît comme la plus complète et relativement facile à ce jour.

La qualité typographique de \LaTeX depuis ses origines a une contrainte : \LaTeX n'est pas WYSIWYG¹ comme par exemple Word, ce qui en rebute plus d'un ! Heureusement, l'utilisation de plus en plus ergonomique des éditeurs de texte pallie en partie à ce qui peut être considéré par certains comme un inconvénient mais par d'autres comme un avantage.

Il faut noter l'existence d'au moins deux éditeurs de texte quasi-WYSIWYG : Texmacs et Lyx. L'auteur n'est pas bien convaincu que ceux-ci apportent une simplification dans l'élaboration d'un document. Ils peuvent cependant être utiles pour l'apprentissage de \LaTeX . Mais rien ne vaut un stage pour bien démarrer, prendre les bonnes habitudes pour devenir ensuite autonome.

À ce jour, 24 mars 2013, seul T \TeX works, éditeur de texte fourni avec T \TeX live et avec MikT \TeX au moment de leur installation, a intégré dans sa barre d'icônes une compilation pour Lua \LaTeX .

Lua \LaTeX semble être appelé à remplacer pdf \LaTeX . Il serait l'avenir de \LaTeX !

Pour résumer :

1. What You See Is What You Get : tel écran, tel écrit , avec Word ou LibreOfficeWriter

moteur :	pdf \LaTeX	X \LaTeX	Lua \LaTeX
fonctionnement avec les polices \LaTeX	OUI	OUI	OUI
fonctionnement avec polices du système d'exploitation	NON	OUI	OUI avec restriction
fonctionnement avec image .ps	NON	OUI	NON
fonctionnement avec image .png	OUI	OUI	OUI
fonctionnement avec PSTricks	NON	OUI	NON
fonctionnement avec TikZ	OUI	OUI	OUI
fonctionnement avec sagetex	OUI	OUI	OUI
langage de programmation intégré	NON	NON	OUI

2 Le préambule d'un document Lua \LaTeX

Il ressemble beaucoup à celui d'un document X \LaTeX . Seul le package xltextra est remplacé par luatextra. Voici un exemple de préambule pour un document ayant un contenu mathématique avec graphique :

```

\documentclass[12pt]{article}
\usepackage{luatextra,xcolor}
\usepackage{tikz}
\usepackage{amsmath}
\usepackage{amsfonts}
\usepackage{amssymb}
\usepackage{unicode-math}
\usepackage[left=1cm,right=1cm,top=1cm,bottom=1.5cm]{geometry}
\usepackage[français]{babel}
\setmainfont[Ligatures=TeX]{Linux Libertine O}
\setmathfont{Asana Math}

```

3 Le langage Lua et sa correspondance en Lua \LaTeX

3.1 Console lua

Pour vérifier l'installation et le fonctionnement de Lua sur l'ordinateur, il est possible d'écrire un script avec quelques lignes de code lua et de les enregistrer dans un fichier <fichier>.lua

Pour que ce fichier soit lu, on peut utiliser par exemple la console, se placer dans le dossier où se trouve le fichier et écrire lua <fichier>.lua pour lancer le programme qui doit alors s'exécuter.

Une autre solution consiste à utiliser un programme dédié à lua pour exécuter le code avec une interface plus conviviale. C'est le cas avec ZeroBrane Studio dont l'installation est très simple sous Windows. Ce logiciel fournit aussi différents exemples de scripts qu'on peut essayer et décortiquer pour apprendre la syntaxe de Lua. Il y a deux possibilités de lancement d'un programme écrit en lua : dans le menu Project, le classique «Run» mais aussi «Run as Scratchpad» qui lance le programme en direct permettant de voir immédiatement l'effet de tout changement dans le code. Notamment, par clic maintenu sur un nombre, en déplaçant la souris, la valeur change avec effet immédiat sur la compilation du document.

3.2 Commentaires en Lua

Ce qui est écrit après `--` est en commentaire

`[[]]` permet des commentaires sur plusieurs lignes.

3.3 Calcul

On utilise la fonction `print()`

3.3.1 Ecriture des nombres

syntaxe lua	syntaxe Lua \LaTeX	résultat
<code>print(12.34*10^-3)</code>	<code>\directlua{tex.print(12.34*10^-3)}</code>	0.01234
<code>print(12.34e-3)</code>	<code>\directlua{tex.print(12.34e-3)}</code>	0.01234
<code>print(12.34E-3)</code>	<code>\directlua{tex.print(12.34E-3)}</code>	0.01234
<code>print(math.pi)</code>	<code>\directlua{tex.print(math.pi)}</code>	3.1415926535898

3.3.2 Arithmétique

syntaxe lua	syntaxe Lua \LaTeX	résultat
<code>print(3+7)</code>	<code>\directlua{tex.print(3+7)}</code>	10
<code>print(3-7)</code>	<code>\directlua{tex.print(3-7)}</code>	-4
<code>print(3*7)</code>	<code>\directlua{tex.print(3*7)}</code>	21
<code>print(3/7)</code>	<code>\directlua{tex.print(3/7)}</code>	0.42857142857143
<code>print(3^7)</code>	<code>\directlua{tex.print(3^7)}</code>	2187

3.3.3 Fonctions trigonométriques

syntaxe lua	syntaxe Lua \LaTeX	résultat
<code>print(math.sin(-3))</code>	<code>\directlua{tex.print(math.sin(-3))}</code>	-0.14112000805987
<code>print(math.cos(-3))</code>	<code>\directlua{tex.print(math.cos(-3))}</code>	-0.98999249660045
<code>print(math.tan(-3))</code>	<code>\directlua{tex.print(math.tan(-3))}</code>	0.14254654307428
<code>print(math.asin(0.5))</code>	<code>\directlua{tex.print(math.asin(0.5))}</code>	0.5235987755983
<code>print(math.acos(0.5))</code>	<code>\directlua{tex.print(math.acos(0.5))}</code>	1.0471975511966
<code>print(math.acos(-3))</code>	<code>\directlua{tex.print(math.acos(-3))}</code>	nan
<code>print(math.atan2(-3,1))</code>	<code>\directlua{tex.print(math.atan2(-3,1))}</code>	-1.2490457723983

3.3.4 Fonctions logarithmiques et exponentielles

syntaxe lua	syntaxe Lua \LaTeX	résultat
<code>print(math.log(2))</code>	<code>\directlua{tex.print(math.log(2))}</code>	0.69314718055995
<code>print(math.log10(100))</code>	<code>\directlua{tex.print(math.log10(100))}</code>	2
<code>print(math.exp(-3))</code>	<code>\directlua{tex.print(math.exp(-3))}</code>	0.049787068367864
<code>print(math.frexp(3))</code>	<code>\directlua{tex.print(math.frexp(3))}</code>	2
<code>print(math.ldexp(0.3,2))</code>	<code>\directlua{tex.print(math.ldexp(0.3,2))}</code>	1.2

3.3.5 Nombres aléatoires

syntaxe lua	syntaxe Lua \LaTeX	résultat
<code>print(math.random())</code>	<code>\directlua{tex.print(math.random())}</code>	0.94077464981972
<code>print(math.random(100))</code>	<code>\directlua{tex.print(math.random(100))}</code>	44
<code>print(math.random(3,10))</code>	<code>\directlua{tex.print(math.random(3,10))}</code>	7
<code>print(math.randomseed(12))</code>	<code>\directlua{tex.print(math.randomseed(12))}</code>	

3.3.6 Autres fonctions mathématiques

syntaxe lua	syntaxe Lua \LaTeX	résultat
<code>print(math.abs(-3))</code>	<code>\directlua{tex.print(math.abs(-3))}</code>	3
<code>print(math.ceil(-3.5))</code>	<code>\directlua{tex.print(math.ceil(-3.5))}</code>	-3
<code>print(math.floor(-3.5))</code>	<code>\directlua{tex.print(math.floor(-3.5))}</code>	-4
<code>print(math.max(-3,1,5.6))</code>	<code>\directlua{tex.print(math.max(-3,1,5.6))}</code>	5.6
<code>print(math.min(-3,1,5.6))</code>	<code>\directlua{tex.print(math.min(-3,1,5.6))}</code>	-3
<code>print(math.mod(8,3))</code>	<code>\directlua{tex.print(math.mod(8,3))}</code>	2
<code>print(math.mod(8,3.2))</code>	<code>\directlua{tex.print(math.mod(8,3.2))}</code>	1.6
<code>print(math.pow(2,0.5))</code>	<code>\directlua{tex.print(math.pow(2,0.5))}</code>	1.4142135623731
<code>print(math.sqrt(3))</code>	<code>\directlua{tex.print(math.sqrt(3))}</code>	1.7320508075689

3.3.7 degrés et radians

syntaxe lua	syntaxe Lua \LaTeX	résultat
<code>print(math.deg(3.14))</code>	<code>\directlua{tex.print(math.deg(3.14))}</code>	179.90874767108
<code>print(math.rad(90))</code>	<code>\directlua{tex.print(math.rad(90))}</code>	1.5707963267949

3.4 Texte et concaténation

syntaxe lua	syntaxe Lua \LaTeX	résultat
<code>print('bonjour')</code>	<code>\directlua{tex.print('bonjour')}</code>	bonjour
<code>print("bonjour")</code>	<code>\directlua{tex.print("bonjour")}</code>	bonjour
<code>print([[bonjour]])</code>	<code>\directlua{tex.print([[bonjour]])}</code>	bonjour
<code>print('bon'..'jour')</code>	<code>\directlua{tex.print('bon'..'jour')}</code>	bonjour

3.5 Boucles

3.5.1 while do end

```
\begin{luacode}
i=3
while i>0 do
tex.print(i)
i=i-1
end
\end{luacode}
```

3 2 1

3.5.2 repeat untill

```
\begin{luacode}
i=3
repeat
tex.print(i)
i=i-1
until i==0
\end{luacode}
```

3 2 1

3.5.3 for do end

```
\begin{luacode}
for i=3,0,-1 do
tex.print(i)
end
\end{luacode}
```

3 2 1 0

Les résultats de la compilation sont mis côte à côte. Pour les avoir les uns en dessous des autres, voici la technique :

```
\begin{luacode}
for i=3,0,-1 do
tex.print(i..'\\')
end
\end{luacode}
```

3
2
1
0

La règle est la suivante : à l'intérieur d'un `tex.print()`,

- une commande \LaTeX `\<commande>` devient `'\\<commande>'`
- `\\` devient `'\\\\'`
- entre une variable — par exemple `i` — et du texte, il faut concaténer avec `..`

4 Exercices

Dans cette section, il s'agit de choisir quelques fonctions mathématiques, de réaliser un tableau de valeurs et de tracer les graphiques correspondants.

4.1 $y = x^2 - 3x + 1$

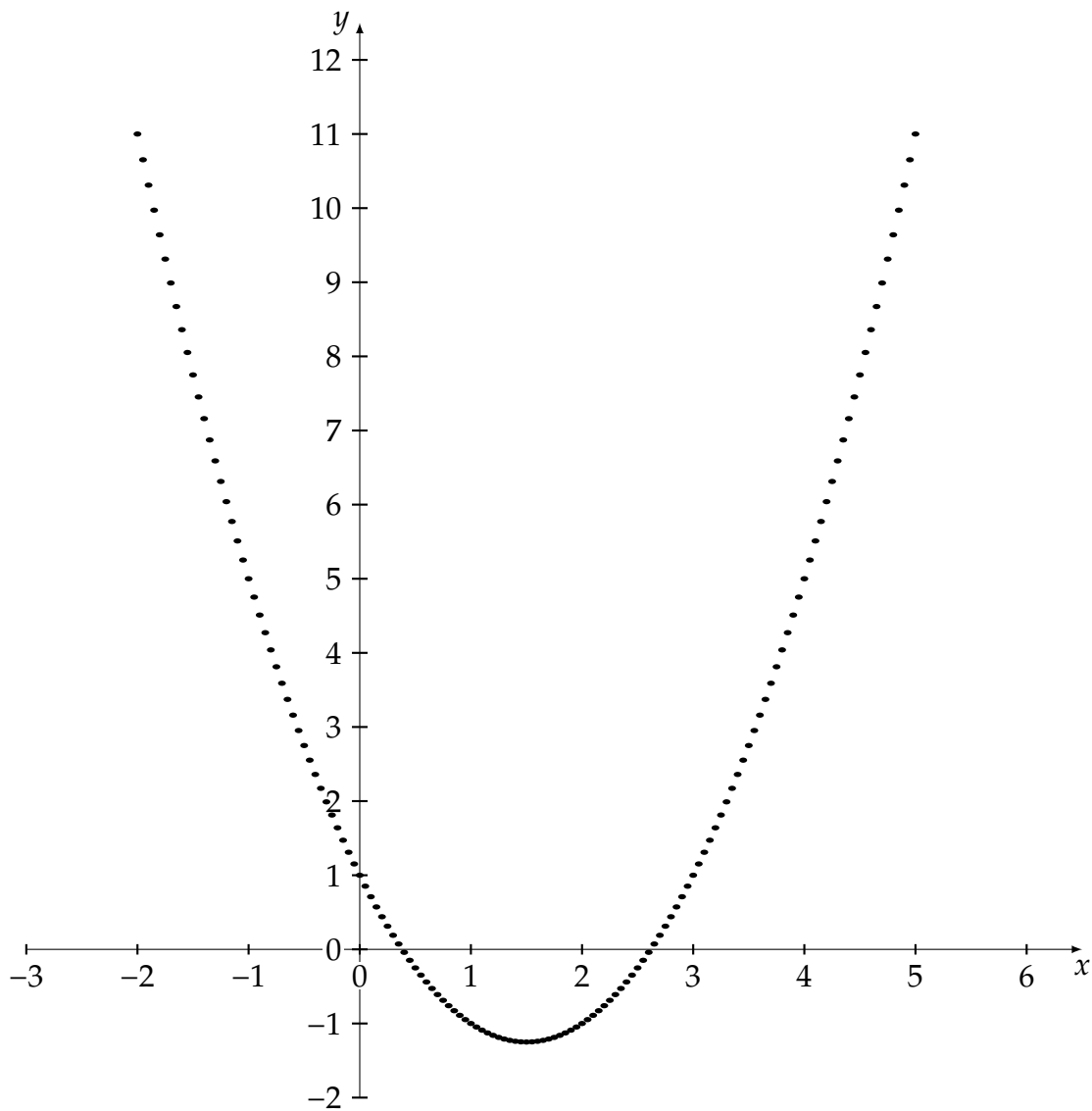
4.1.1 Tableau de valeurs

```
\begin{luacode}
tex.print("\\begin{tabular}{c|c}")
tex.print("x & y \\\\ \\hline")
for x=-2,5,0.5 do
tex.print( x .. '&' .. x^2-3*x+1 .. '\\ \\ ' )
end
tex.print("\\end{tabular}")
\end{luacode}
```

x	y
-2	11
-1.5	7.75
-1	5
-0.5	2.75
0	1
0.5	-0.25
1	-1
1.5	-1.25
2	-1
2.5	-0.25
3	1
3.5	2.75
4	5
4.5	7.75
5	11

4.1.2 Graphe

```
\begin{tikzpicture}[xscale=1.5]
\tkzInit[xmin = -3,xmax =6,ymin = -2,ymax = 12]
\tkzAxeXY
\begin{luacode}
for x=-2,5,0.05 do
tex.print("\fill(".. x .."," .. x^2-3*x+1 ..")circle (1pt);")
end
\end{luacode}
\end{tikzpicture}
```



x	y
-2	0.090909090909091
-1.5	0.12903225806452
-1	0.2
-0.5	0.36363636363636
0	1
0.5	-4
1	-1
1.5	-0.8
2	-1
2.5	-4
3	1
3.5	0.36363636363636
4	0.2
4.5	0.12903225806452
5	0.090909090909091

