

Présenter des algorithmes, morceaux de programmes avec LaTeX

LaTeX est un système de composition de document très adapté aux documents scientifiques. Cet article s'adresse donc à des personnes connaissant déjà LaTeX, et désirant rédiger des algorithmes avec LaTeX. Mais, c'est aussi l'occasion pour ceux qui ne connaissent pas d'avoir un aperçu des possibilités de ce système. L'IREM de Lyon a d'ailleurs produit une brochure spécialement dédié aux enseignants de collège et de lycée pour apprendre à utiliser LaTeX¹.

L'algorithmique et la programmation sont de plus en plus présente dans les programmes de Mathématiques, au collège et au lycée². On présente ci-dessous différents outils pour rédiger des algorithmes dans LaTeX afin de préparer des activités, des feuilles d'exercices, des sujets de contrôles ou d'examens de niveau collège ou lycée.

1 Pourquoi utiliser LaTeX ?

LaTeX permet de produire des documents de très haute qualité. On pourra comparer par exemple deux versions du sujet de Brevet Centre Étranger 2017, en particulier les courbes de l'exercice 2. Dans l'un³, la copie d'écran des graphiques fait apparaître des signes « moins ». Dans l'autre⁴, réalisé avec LaTeX, les graphiques sont correctement réalisés.

Un des avantages, de LaTeX par rapport à Word ou LibreOffice est la possibilité de dessiner directement dans LaTeX, sans passer par un autre logiciel. Prenons Scratch comme exemple. Pour rédiger une évaluation, on peut faire le programme avec Scratch, puis faire une copie d'écran et l'insérer dans Word. Le problème est que le rendu à l'impression en noir et blanc n'est pas terrible (voir par exemple le document d'accompagnement d'Eduscol). Cela pose des problèmes de lecture pour les élèves, en particulier les élèves à besoins particuliers. La méthode utilisant LaTeX permet d'obtenir une image de meilleur qualité, car elle est redessinée. L'argument est valable pour d'autres illustrations comme des représentations graphiques de fonctions ou de graphes.

Un autre avantage est de pouvoir modifier certains éléments du dessin (taille, couleur, police par exemple). Voici quatre variations pour un même script Scratch :

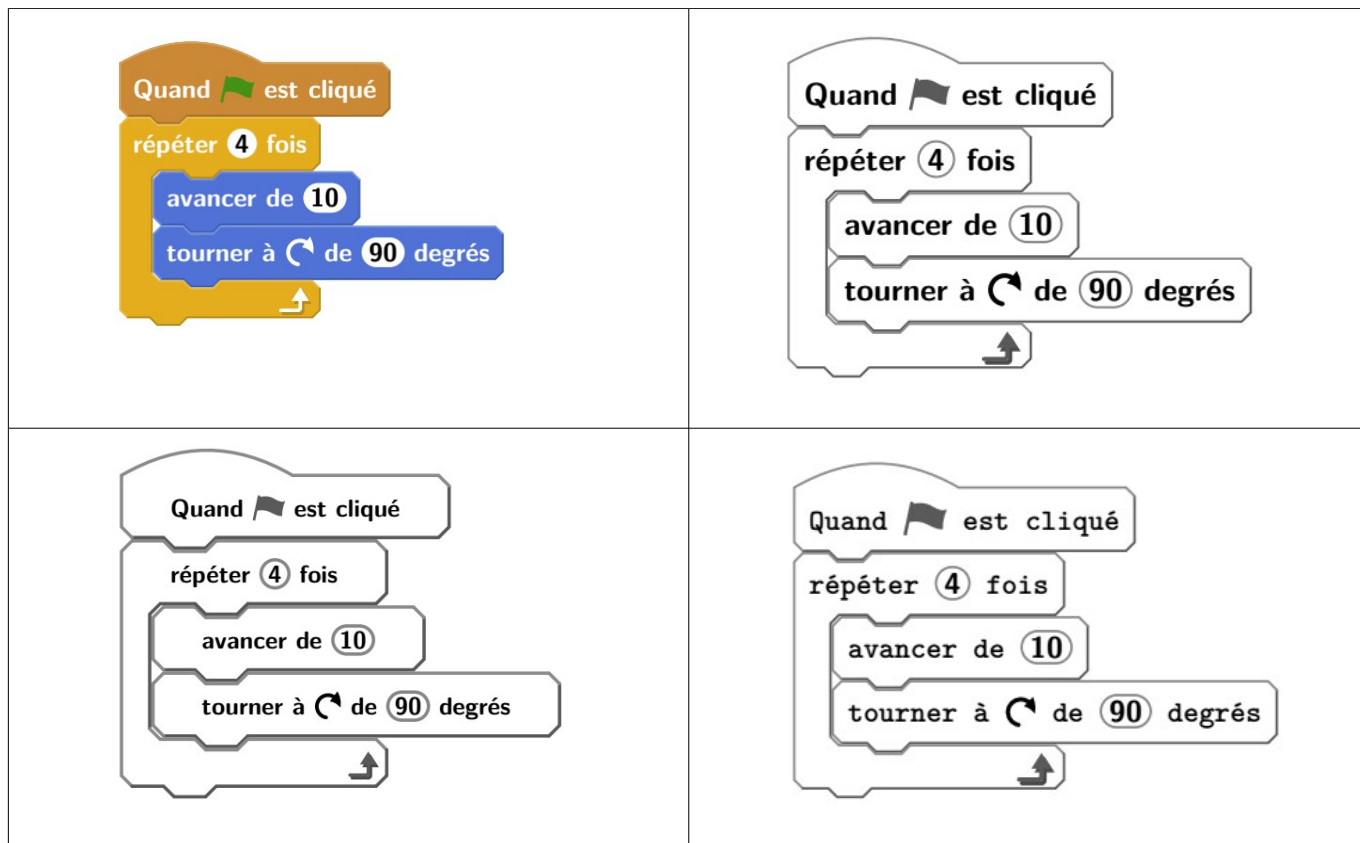
1 <http://math.univ-lyon1.fr/irem/spip.php?article340>

2 Eduscol :

https://cache.media.eduscol.education.fr/file/Mathematiques/73/3/Algorithmique_et_programmation_787733.pdf

3 <http://www.cafepedagogique.net/lexpresso/Documents/DNB2017/DNB2017MATHSERIEGENERALE.pdf>

4 https://www.apmep.fr/IMG/pdf/brevet-CE-19_juin-2017.pdf



Enfin, pour sortir du cadre de l'algorithmique, notons que les blocs Texte de Geogebra peuvent être écrit en LaTeX, et que les figures de Geogebra peuvent être exportés en code lisible par LaTeX.

Cinq langages d'algorithmique ou de programmation sont présentés :

- le pseudo-code
- Scratch
- Geotortue
- Algobox
- Python

Toutes les méthodes présentées se basent sur des packages déjà existants (un package est une extension de LaTeX permettant de réaliser des tâches particulières). Il n'y a que pour Geotortue où il a fallu utiliser les capacités de personnalisation d'un package.

Enfin, quelques commentaires sur les langages présentés :

- Ce sont ceux que j'utilise en classe !
- Il manque les programmes de type Blockly ou Sofus, et les algorithmes présentés sous forme de diagramme.
- Une nouvelle version de Scratch devrait sortir au premier trimestre 2018, mais la présentation devrait rester quasiment identique.
- Il y a une évolution de l'écriture des algorithmes pour le baccalauréat⁵.

⁵ <http://disciplines.ac-montpellier.fr/mathematiques/examens-et-evaluation/bac-2018>

Pour chaque langage ou logiciels, on présente ce qui se rapproche d'un ECM (Exemple Complet Minimal). Ces exemples ont été testés avec Texmaker⁶ et Miktex⁷.

2 Le pseudo-code

Une autre méthode a déjà été décrite dans Mathématique par Alain Busser⁸. Ici, nous utilisons le package algorithm2e⁹.

2.1 Le code LaTeX

```
\documentclass[a4paper,10pt]{article}
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{lmodern}
\usepackage[frenchb]{babel}
\usepackage[linesnumbered, french]{algorithm2e}

\begin{document}

\begin{algorithm}
$n \leftarrow 0$ \;
$u \leftarrow 3081,45$ \;
\Tq{$u < k$}{
$u \leftarrow 1,04 \times u$ \;
$n \leftarrow n+1$ \;
}
\end{algorithm}

\end{document}
```

2.2 Le résultat

```
1  $n \leftarrow 0$  ;
2  $u \leftarrow 3081,45$  ;
3 tant que  $u < k$  faire
4   |  $u \leftarrow 1,04 \times u$  ;
5   |  $n \leftarrow n + 1$  ;
6 fin
```

6 http://www.xm1math.net/texmaker/index_fr.html

7 <https://miktex.org/>

8 Source Alain Busser : <http://revue.sesamath.net/spip.php?article295>

9 Source sur le CTAN : <https://ctan.org/pkg/algorithm2e>

3 Scratch

Le package Scratch¹⁰ dessine des blocs ressemblant à ceux du logiciel Scratch. Une option du package permet de dessiner en noir et blanc (par défaut), en couleur ou en niveaux de gris. La documentation est en français. C'est ce package qui est utilisé dans les annales du brevet de l'APMEP.

3.1 Le code LaTeX

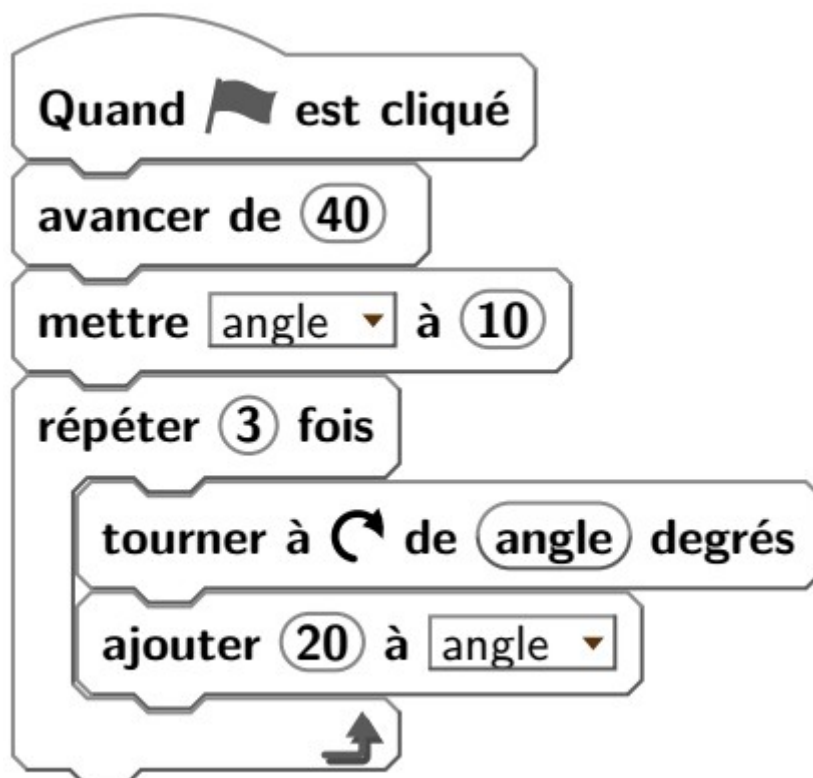
```
\documentclass[a4paper,10pt]{article}
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{lmodern}
\usepackage[frenchb]{babel}
\usepackage{scratch}

\begin{document}

\begin{scratch}
\blockinit{Quand \greenflag est cliqué}
\blockmove{avancer de \ovalnum{40} }
\blockvariable{mettre \selectmenu{angle} à \ovalnum{10}}
\blockrepeat{répéter \ovalnum{3} fois}
{
\blockmove{tourner à \turnright{}} de \ovalvariable{angle} degrés}
\blockvariable{ajouter \ovalnum{20} à \selectmenu{angle}}
}
\end{scratch}

\end{document}
```

3.2 Le résultat



10 Source sur le CTAN : <https://ctan.org/pkg/scratch>

4 Geotortue

Il n'existe pas de package spécifique à Geotortue. Je propose une solution qui est d'utiliser et de personnaliser le package listings¹¹. Le package listings a des problèmes avec certains caractères. Une solution est proposé dans la partie Python qui utilise le même package.

4.1 Le code LaTeX

```
\documentclass[a4paper,10pt]{article}
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{lmodern}
\usepackage[frenchb]{babel}

\usepackage{xcolor}

\definecolor{geoKeyword}{RGB}{153,0,51}
\definecolor{geoCommand}{RGB}{0,0,123}
\definecolor{geoNumbercolor}{RGB}{181,181,181}
\definecolor{geoText}{RGB}{255,0,192}
\definecolor{geoFonction}{RGB}{51,51,51}

\usepackage{listings}

\lstdefinlanguage{Geotortue}{
morekeywords={rep, remplis, si, tant_que, boucle, écris, à, aff, def, eff, init,
pause, mg, photo, retiens, retourne, pour, fin, dis},
morekeywords=[2]{av, re, td, tg, vg, ct, mt, lc, bc, crayon, palette, pvd, pvg, pvh,
pvb, miroir, tlp, boussole, vise, imite, pvxy, pvxz, pvyz},
morekeywords=[3]{sqrt, abs, reste, sin, cos, tan, asin, acos, atan, sinh, cosh, tanh,
asinh, acosh, atanh, ln, log, round, floor, binom, rand, alea},
morestring=[b][\color{geoText}]",
}
\renewcommand*\thelstnumber{\the\value{lstnumber}>}
\lstdefinestyle{numbers}{
numbers=left,
stepnumber=1,
numberstyle=\tiny\ttfamily\color{geoNumbercolor},
xleftmargin=2em,
numbersep=2pt,
}
\lstdefinestyle{mygeotortue}{
language=Geotortue,
basicstyle=\ttfamily,
keywordstyle={\color{geoKeyword}\bfseries},
keywordstyle=[2]{\color{geoCommand}\bfseries},
keywordstyle=[3]{\color{geoFonction}\bfseries},
}

\begin{document}

\begin{lstlisting}[language=Geotortue, style=mygeotortue, style=numbers]
pour carre
rep 2 [av 8 ; td 90 ; av sqrt(64) ; td 90]
fin
\end{lstlisting}

\end{document}
```

11 Source sur le CTAN : <https://ctan.org/pkg/listings>

4.2 Le résultat

```
1> pour carre
2> rep 2 [av 8 ; td 90 ; av sqrt(64) ; td 90]
3> fin
```

5 Algobox

Depuis le début du mois de septembre 2017, un package algobox¹² pour LaTeX est disponible. Un export depuis le logiciel Algobox était déjà possible, mais il fournissait un code assez long. Avec le package, il suffit de copier-coller dans l'éditeur depuis le logiciel Algobox, puis de modifier un peu le code.

5.1 Le code LaTeX

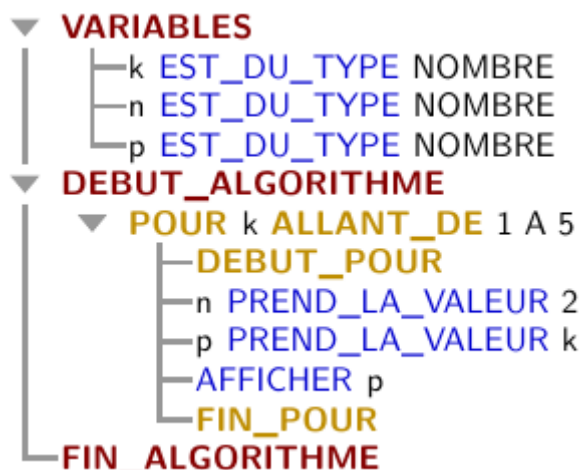
```
\documentclass[a4paper,10pt]{article}
\usepackage[utf8]{inputenc}% ou \usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{lmodern}
\usepackage[frenchb]{babel}
\usepackage{algobox}

\begin{document}

\begin{algobox}
\; \; \; \VARIABLES
\; \; \; k \ESTDUYPE NOMBRE
\; \; \; n \ESTDUYPE NOMBRE
\; \; \; p \ESTDUYPE NOMBRE
\; \; \; \DEBUTALGORITHME
\; \; \; \POUR k \ALLANTDE 1 \A 5
\; \; \; \DEBUTPOUR
\; \; \; n \PRENDLAVALEUR 2*k
\; \; \; p \PRENDLAVALEUR k*n-4
\; \; \; \AFFICHER p
\; \; \; \FINPOUR
\; \; \; \FINALGORITHME
\end{algobox}

\end{document}
```

5.2 Le résultat



12 Source sur le CTAN : <https://ctan.org/pkg/algobox>

6 Python

Je ne donne qu'un petit exemple car il est assez facile de trouver des méthodes. Le package listings ne gère pas très bien l'encodage UTF8 ce qui explique la nécessité de la dizaine de lignes contenant les caractères accentués.

Cette façon de rédiger est valable pour la plupart des langages de programmation car le package listings connaît des dizaines de langages (par exemple Caml, Matlab, HTML, Ruby...)

6.1 Le code LaTeX

```
\documentclass[a4paper,10pt]{article}
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{lmodern}
\usepackage[frenchb]{babel}

\usepackage{xcolor}
\usepackage{listings}
\lstset{
  literate=
  {á}{\`a}}1 {é}{\`e}}1 {í}{\`i}}1 {ó}{\`o}}1 {ú}{\`u}}1
  {Á}{\`A}}1 {É}{\`E}}1 {Í}{\`I}}1 {Ó}{\`O}}1 {Ú}{\`U}}1
  {à}{\`a}}1 {è}{\`e}}1 {ì}{\`i}}1 {ò}{\`o}}1 {ù}{\`u}}1
  {À}{\`A}}1 {È}{\`E}}1 {Ì}{\`I}}1 {Ò}{\`O}}1 {Ù}{\`U}}1
  {ä}{\`"a}}1 {ë}{\`"e}}1 {ï}{\`"i}}1 {ö}{\`"o}}1 {ü}{\`"u}}1
  {Ä}{\`"A}}1 {Ë}{\`"E}}1 {Ï}{\`"I}}1 {Ö}{\`"O}}1 {Ü}{\`"U}}1
  {â}{\`^a}}1 {ê}{\`^e}}1 {î}{\`^i}}1 {ô}{\`^o}}1 {û}{\`^u}}1
  {Â}{\`^A}}1 {Ê}{\`^E}}1 {Î}{\`^I}}1 {Ô}{\`^O}}1 {Û}{\`^U}}1
  {œ}{\`oe}}1 {Œ}{\`OE}}1 {æ}{\`ae}}1 {Æ}{\`AE}}1 {ß}{\`ss}}1
  {ú}{\`H{u}}1 {Ú}{\`H{U}}1 {ó}{\`H{o}}1 {Ó}{\`H{O}}1
  {ç}{\`c c}}1 {Ç}{\`c C}}1 {ø}{\`o}}1 {â}{\`r a}}1 {Á}{\`r A}}1
  {€}{\`EUR}}1 {£}{\`pounds}}1
}
\lstdefinestyle{stylepython}{
  language=Python,
  basicstyle=\ttfamily,
  commentstyle=\color{green},
  keywordstyle=\color{blue},
  stringstyle=\color{olive},
  numberstyle=\tiny,
  numbers=left,
  stepnumber=1,
  numbersep=5pt
}

\begin{document}

\begin{lstlisting}[style=stylepython]
#convertir un triplet rgb en une chaîne hexadécimale
def rgb2hex(red, green, blue):
    return '#%02x%02x%02x' % (red, green, blue)

ma_couleur=rgb2hex(255,18,0)
\end{lstlisting}

\end{document}
```


6.2 Le résultat

```
1 #convertir un triplet rgb en une chaîne hexadécimale
2 def rgb2hex(red, green, blue):
3     return '#%02x%02x%02x' % (red, green, blue)
4
5 ma_couleur=rgb2hex(255,18,0)
```

7 Conclusion

J'espère que cet article donnera l'envie à certains de commencer à composer des documents avec LaTeX. Vous trouverez en fin d'article un exemple de ce qu'il est possible de faire avec LaTeX dans un petit feuillet de 9 pages contenant des exercices. Vous pouvez aussi regarder le site Chingatome qui regroupe des centaines d'exercices rédigés en LaTeX. On trouve une présentation de ce site dans le numéro 53 de la revue *Mathematice*¹³.

13 <http://revue.sesamath.net/spip.php?article912>