

Recherche d'imperfections typographiques avec LuaLaTeX

Daniel Flipo

daniel.flipo@free.fr

1 De quoi s'agit-il?

L'extension `lua-tylo` décrite ci-dessous¹ permet de mettre en lumière par un changement de couleur, les lignes typographiquement imparfaites d'un fichier PDF produit par LuaLaTeX. Une liste des pages concernées est affichée à la fin du fichier `.log`, permettant un accès rapide aux pages potentiellement défectueuses.

Normalement, c'est-à-dire lorsque la justification n'est pas trop étroite, (Lua)TeX fait du bon travail, mais il peut rester des points à vérifier, notamment des lignes trop pleines ou lavées (*Overfull*, *Underfull box*), des veuves et des orphelines, des mots coupés en fin de page ou d'alinéa ou sur plusieurs lignes consécutives, des dernières lignes d'alinéa trop courtes ou presque pleines, des pages quasi vides. La répétition d'un même mot ou partie de mot au début ou à la fin de deux lignes consécutives est aussi détectée. La présence en fin de ligne de certains mots très courts (une ou deux lettres, liste dépendant de la langue) peut également être recherchée.

`lua-tylo`, ne fonctionne qu'avec LuaLaTeX et *ne corrige aucun des défauts relevés* mais se contente de *signaler* au relecteur les points qui peuvent nécessiter une correction. Seul un humain entraîné peut décider si une ligne légèrement lavée est acceptable ou non, ou si la suppression d'une coupure malvenue ne va pas provoquer des désordres plus graves encore.

Un outil potentiellement utile est la commande TeX `\spaceskip` qui, utilisée dans un groupe, permet de modifier localement l'espace inter-mots. Un léger accroissement de celui-ci dans un paragraphe peut rendre acceptable la dernière ligne de celui-ci alors qu'elle était trop courte ou, si elle était presque pleine, lui ajouter une ligne, supprimant ainsi une orpheline éventuelle. De même une légère réduction de l'espace inter-mots peut supprimer la dernière ligne (courte) d'un paragraphe et éviter une veuve en début de la page suivante.

Je conseille de n'appliquer `lua-tylo` que sur des textes « presque au point », d'améliorer ce qui peut l'être puis de *supprimer* l'appel à `lua-tylo` afin de ne pas risquer de mettre en lumière les imperfections que l'on aura renoncé à corriger. Pour appliquer toutes les vérifications prévues par `lua-tylo`, il suffit d'ajouter dans le préambule la ligne `\usepackage[All]{lua-tylo}`

La version courante (0.60) nécessite un noyau LaTeX récent, 2021/06/01 ou ultérieur. Ceux qui ne disposent que d'un noyau plus ancien reçoivent un message d'avertissement et un message d'erreur « `Unable to register callback` » ; une version « rollback » est prévue à leur intention, elle se charge par la commande `\usepackage[All]{lua-tylo}[=v0.4]`.

Les fichiers `demo.tex` et `demo.pdf` fournissent un exemple du traitement opéré par `lua-tylo`.

1. Version v.0.60, mise à jour le 2023-02-04.

Un grand merci à Jacques André et Thomas Savary pour avoir accepté de tester les pré-versions et pour leurs retours riches et toujours pertinents ; leurs suggestions et leurs encouragements ont grandement contribué à améliorer la première version mise en ligne.

2 Utilisation

Comme indiqué plus haut la vérification la plus complète s'obtient par :

```
\usepackage[All]{lua-typo}
```

Il est possible de choisir les tests à activer de deux manières, soit « tout sauf ... » soit « seulement ceci et cela ». Pour tout activer sauf les options `<OptX>` et `<OptY>` :

```
\usepackage[All, <OptX>=false, <OptY>=false]{lua-typo}
```

ou pour se limiter aux tests `<OptX>` et `<OptY>` :

```
\usepackage[<OptX>, <OptY>, <OptZ>]{lua-typo}
```

Le tableau suivant donne le nom des options et le type des vérifications proposées :

Nom	Imperfection à signaler
All	Active toutes les options ci-dessous
ShortLines	Dernière ligne d'alinéa trop courte?
BackParindent	Dernière ligne d'alinéa <i>presque</i> pleine?
ShortPages	Page quasi vide (quelques lignes)?
OverfullLines	Ligne trop pleine?
UnderfullLines	Ligne lavée?
Widows	Veuve (haut de page)?
Orphans	Orpheline (bas de page)?
EOPHyphens	Mot coupé en bas de page?
RepeatedHyphens	Coupsures sur trop de lignes consécutives?
ParLastHyphen	Coupsure à l'avant-dernière ligne d'un alinéa?
EOLShortWords	Mots courts (1 or 2 lettres) en fin de ligne?
FirstWordMatch	Même (partie de) mot en début de lignes consécutives?
LastWordMatch	Même (partie de) mot en fin de lignes consécutives?
FootnoteSplit	Fin de note de bas de page sur page suivante?

Par exemple, pour limiter les vérifications aux lignes trop pleines ou creuses, il suffit de coder :

```
\usepackage[OverfullLines, UnderfullLines]{lua-typo}
```

Pour tout vérifier sauf les coupures répétées en fin de ligne on codera :

```
\usepackage[All, RepeatedHyphens=false]{lua-typo}
```

Notez que l'option `that All` doit être la première de la liste, les suivantes étant retirées de la liste complète définie par `All`.

Le nom des différentes options n'étant pas facile à mémoriser, il est possible de les retrouver sans devoir consulter la documentation ; l'option `ShowOptions` affiche la liste complète dans le fichier `.log` : `\usepackage[ShowOptions]{lua-typo}`

L'option `None`, empêche toute vérification : `\usepackage[None]{lua-typo}` a pour effet de supprimer complètement tout ajout de code LuaTeX (aucune fonction n'est ajoutée aux *callbacks* de LuaTeX). Cette option peut-être utile lors de la toute dernière compilation, elle n'est pas tout-à-fait équivalente à la mise en commentaire de la ligne

car les variables utilisées par `luatypo` restent définies ; si certaines ont été modifiées dans le préambule aucun message d'erreur du type “*Undefined Control Sequence*” ne sera émis à leur sujet.

Terminons par quelques précisions sur ces options.

FirstWordMatch : les répétitions en début de ligne dans les listes ne sont pas signalées. Ceci est voulu car elles résultent d'un choix délibéré de l'auteur.

LastWordMatch : le dernier mot d'un alinéa qui se termine à plus de `1em` (en fait `\luatypoBackPI`) de la marge droite n'est jamais signalé comme répétition, même s'il est identique au mot de la ligne du dessus. De même, s'il est identique au dernier mot de la ligne suivante, la répétition ne sera pas signalée non plus.

ShortPages : lorsque le nombre de lignes d'une page est jugé insuffisant (voir ci-dessous), seule la dernière ligne de celle-ci est mise en couleur.

RepeatedHyphens : de même, lorsque le nombre de lignes consécutives affectées par des coupures dépasse le seuil fixé (voir ci-dessous), ne sont colorisées que les coupures en excès.

À partir de la version 0.50, le contenu des notes de bas de pages est également vérifié, les notes trop longues pour se terminer sur la page de leur appel sont signalées (option **FootnoteSplit**). Un fichier de suffixe `.typo` est créé, il contient la liste complète des défauts repérés avec leur localisation dans le fichier de sortie.

3 Paramétrage personnalisé

Pour certaines vérifications faites par `lua-typo` un paramétrage est nécessaire : à partir de quelle limite une dernière ligne d'alinéa est-elle considérée comme trop courte ? Combien de coupures consécutives en bout de ligne sont-elles acceptables ? Ces réglages dépendent évidemment du contexte, un correcteur de romans aura des exigences plus strictes qu'un auteur de documentation technique par exemple...

`lua-typo` permet de modifier le réglage des curseurs soit dans le fichier `lua-typo.cfg` soit dans le préambule après l'appel de `lua-typo` ; les réglages placés dans le préambule prévalent sur ceux du fichier `lua-typo.cfg` qui eux-mêmes prévalent sur les réglages internes de l'extension.

Le fichier `lua-typo.cfg` fourni avec la distribution reprend exactement les réglages internes, il se trouve normalement dans le répertoire `TEXMFDIST` des distributions TeXLive, MikTeX, etc. L'utilisateur a la possibilité de recopier ce fichier soit dans son répertoire de travail pour un document particulier, soit dans son répertoire `TEXMFHOME` ou `TEXMFLOCAL` et de le personnaliser comme il l'entend.

Voici la liste complète des paramètres personnalisables avec leur valeur par défaut, leurs noms sont systématiquement préfixés par `luatypo` afin d'éviter de possibles conflits avec d'autres extensions.

BackParindent : la dernière ligne d'un alinéa fixe devrait, soit être pleine — en fait se terminer à moins de `\luatypoBackFuzz=2pt` de la marge droite —, soit s'en éloigner d'au moins de `\luatypoBackPI=1em`.

ShortLines : `\luatypoLLminWD=2\parindent`² fixe la longueur minimale acceptable pour la dernière ligne d'un alinéa.

2. Ou `20pt` si `\parindent=0pt`.

ShortPages : `\luatypoPageMin=5` fixe le nombre minimal de lignes d'une page pour que celle-ci ne soit pas déclarée trop courte. En fait, la position de la dernière ligne est prise en compte afin que les pages de titre ou celles contenant une image ne soient pas signalées comme fautives.

RepeatedHyphens : `\luatypoHyphMax=2` fixe le nombre maximal acceptable de lignes consécutives terminées par un mot coupé.

UnderfullLines : `\luatypoStretchMax=200` fixe le pourcentage maximal acceptable pour l'étirement des espaces-mots, au-delà la ligne est déclarée lavée. La valeur donnée doit être un entier supérieur ou égal à 100, cette valeur 100 correspond à l'étirement maximal prévu par la fonte (`\fontdimen3`); avec ce réglage attendez-vous à trouver une kyrielle de lignes les creuses! En fait la valeur par défaut (200) correspond approximativement à ce que TeX, avec les réglages par défaut (`\tolerance=200`, `\hbadness=1000`), considère comme *Underfull hbox*.

First/LastWordMatch: `\luatypoMinFull=3` et `\luatypoMinPart=4` nombres minimaux de lettres identiques (resp. pour un mot complet ou pour une partie de mot) au début ou à la fin de deux lignes consécutives déclenchant l'avertissement. Avec ce réglage (3 et 4), seront détectées deux lignes se terminant par « cible » et « invincible » (quatre lettres en commun), ainsi que la présence de « mon » en début ou fin de deux lignes consécutives (trois lettres en commun).

EOLShortWords: cette option signale la présence en fin de ligne de mots très courts (une ou deux lettres) qui sont répertoriés dans une des listes suivantes (elles dépendent de la langue courante) :

```
\luatypoOneChar{<langue>}{<liste de mots>}
\luatypoTwoChars{<langue>}{<liste de mots>}
```

Lorsque les listes correspondant à la langue du document sont vides, aucune vérification n'est effectuée. Pour l'instant, il y a deux lignes (non actives) prévues pour le français :

```
\luatypoOneChar{french}{'À à Ô'}
\luatypoTwoChars{french}{'Je Tu Il On'}
```

Deux contraintes sont à respecter lorsqu'on veut personnaliser ces listes :

a) le premier argument (langue) *doit être connu de babel*, aussi les commandes `\luatypoOneChar` et `\luatypoTwoChars`, si elles sont utilisées, doivent l'être *après* le chargement de `babel`, une bonne habitude à prendre est donc de toujours charger `lua-typo` *après* `babel`; b) le second argument *doit être une chaîne de caractères*, donc entourée de simples ou doubles *quotes* ASCII et composées de mots séparés par des espaces comme dans les exemples ci-dessus.

À chacune des vérifications faites par `lua-typo` peut être attachée une couleur spécifique pour mettre en évidence les imperfections détectées. Actuellement, seulement cinq couleurs sont utilisées par défaut, voici leur définition dans `lua-typo.cfg` :

```
% \definecolor{mygrey}{gray}{0.6}
% \definecolor{myred}{rgb}{1,0.55,0}
% \luatypoSetColor0{red}      % Coupure à l'avant-dernière ligne
% \luatypoSetColor1{red}     % Coupure en bas de page
% \luatypoSetColor2{red}     % Coupures consécutives
% \luatypoSetColor3{red}     % Mot court en fin de ligne
```

```

% \luatypoSetColor4{cyan}      % Veuve
% \luatypoSetColor5{cyan}      % Orpheline
% \luatypoSetColor6{cyan}      % Dernière ligne d'alinéa trop courte
% \luatypoSetColor7{blue}      % Ligne trop pleine
% \luatypoSetColor8{blue}      % Ligne creuse
% \luatypoSetColor9{red}       % Page presque vide (qq. lignes)
% \luatypoSetColor{10}{myred}  % Répétitions en début de ligne
% \luatypoSetColor{11}{myred}  % Répétitions en fin de ligne
% \luatypoSetColor{12}{mygrey}% Dernière ligne d'alinéa presque pleine
% \luatypoSetColor{13}{cyan}   % Note de bas de page éclatée
%
```

`lua-typo` charge l'extension graphique `color`. Seules les couleurs portant un nom (*named colors*) peuvent être utilisées; pour en définir de nouvelles, il faut donc soit utiliser la commande `\definecolor` de l'extension `color` (comme ci-dessus pour `mygrey` ou `myred`), soit charger l'extension `xcolor` package qui donne accès à une kyrielle de noms de couleurs.