

METTRE EN PAGES UN ROMAN AVEC LA FAMILLE TEX ?

Bien sûr, il y a moins performant, mais c'est plus cher...

par Thomas SAVARY — Révision de mars 2020

© Licence Creative Commons CC BY-SA 4.0. Attribution – partage dans les mêmes conditions 4.0 International (CC BY-SA 4.0).

« Le discours convenu sur la démocratisation du livre comme synonyme d'inévitable relâchement esthétique est irrecevable. » — Olivier BESSARD-BANQUY.

MISE EN PAGE et composition sont au service du texte et du lecteur. La première consiste à définir la disposition graphique, dans l'espace de la page, de contenus informationnels (texte, images, tableaux, graphiques...); la seconde, à assembler les caractères pour former des lignes de texte. L'espacement régulier des mots est la première qualité d'une bonne composition typographique. L'un des pires défauts serait de voir sur une même page alterner des lignes à la composition serrée avec d'autres à la composition relâchée. Un typographe consciencieux veille par ailleurs à éviter les « lézardes », « rues » et « cheminées » (alignement malvenu d'espaces les unes sous les autres ¹ qui donne l'impression de trouées blanches dans le texte, zigzagantes, obliques ou verticales); il aura également à cœur d'éviter certaines fins de ligne disgracieuses, ou encore de s'interroger sur la fin de chaque « belle page » (à droite ²), en vue d'atténuer autant que faire se peut la rupture créée par la nécessité de tourner la page.

Avec les progrès techniques, on aurait pu croire que les ouvrages édités aujourd'hui surpasseraient ceux d'autrefois. Il n'en est rien, généralement. Comment expliquer par exemple que de simples romans de gare des années 1960 aient souvent été mieux composés que les Goncourt de notre époque ? La publication assistée par ordinateur (PAO) n'allait-elle pourtant pas nous faire gagner du temps, pour un travail de qualité au moins égale à celle dont étaient capables nos ancêtres ? Bien sûr, elle le peut. Et si elle le peut, *elle le doit*. Encore faut-il savoir se servir des logiciels de PAO et surtout quoi en faire...

MARCHÉ FRANÇAIS DE LA PAO EN 2020

Chez les éditeurs français (en interne) ou chez les prestataires indépendants (TPE et PME, entrepreneurs individuels), le marché de la PAO est aujourd'hui dominé par Adobe InDesign, après que dans le courant des années 2000 celui-ci a détrôné QuarkXPress, toujours utilisé dans les grandes entreprises, aux côtés de solutions plus coûteuses encore comme le champion de l'automatisation

1. En typographie, le nom *espace* est féminin.

2. Elle est également appelée « recto ». Le verso, on l'aura deviné, n'est autre que la page de gauche, aussi appelée « fausse page ».

Arbortext Advanced Print Publisher (anciennement 3B2), utilisé par exemple par la société IGS-CP.

Après l'ajout de nouvelles fonctionnalités et une baisse de prix conséquente, QuarkXPress semble toutefois déterminé à revenir dans la course s'agissant des PME et des graphistes indépendants. Depuis juin 2019 et la sortie de la première version stable d'Affinity Publisher, il faut toutefois compter aussi avec la société britannique Serif Europe, bien décidée à empiéter sur les plates-bandes d'Adobe avec un prix de vente très agressif de quelques dizaines d'euros. Bientôt la fin de l'hégémonie d'InDesign ?

Affinity Publisher, XPress ou InDesign sont des logiciels polyvalents qui permettent de concevoir des documents aussi variés qu'affiches, dépliants, brochures, magazines, journaux, beaux livres, ouvrages techniques, et bien sûr romans ou essais. Pour le professionnel qui serait ainsi amené à produire toutes sortes de documents, le choix de l'un ou l'autre de ces programmes semble s'imposer. Sont-ils pour autant les plus adaptés à l'édition littéraire ?

Avant de répondre bien évidemment par la négative à cette question, il paraît important de préciser certains points.

L'informatique est censée libérer l'être humain des tâches mécaniques et répétitives en mettant les capacités de calcul des ordinateurs au service de la recherche des solutions optimales, pour accroître à la fois la qualité et l'efficacité. Il ne fait aucun doute d'une part que des logiciels comme InDesign, QuarkXPress ou le jeune Affinity Publisher sont capables de produire des documents d'excellente facture, d'autre part que l'on obtiendra toujours de meilleurs résultats avec un outil que l'on maîtrise qu'avec un autre plus performant dont on ne sait guère se servir.

La question ne s'en pose pas moins de savoir s'il existerait des logiciels capables à moindre coût de performances aux moins égales à celles des trois principaux logiciels se disputant le marché. La réponse cette fois est bien évidemment oui... mais cela dépend...

PLUS PERFORMANT QU'INDESIGN OU XPRESS ?

On aurait pu évoquer un programme de PAO comparable aux trois derniers logiciels mentionnés, en ceci qu'il consiste là encore en un outil polyvalent à l'interface graphique similaire. Il s'agit bien entendu de Scribus. Ce n'est pourtant pas de lui qu'il va s'agir : le fait d'être un logiciel libre et gratuit constitue certes un avantage économique et éthique, mais ce programme ne semble pas aussi complet et performant à ce jour que ses concurrents, même si de fait certains professionnels l'utilisent depuis quelques années. Nous allons parler ici d'une autre famille de logiciels, proposant une approche radicalement différente : *les programmes de la famille TeX*³.

QuarkXPress, InDesign, Affinity Publisher ou Scribus sont des programmes de type WYSIWYG (*what you see is what you get* — « vous obtiendrez ce que vous voyez »). Rien de tel s'agissant des outils de la famille TeX. Avec eux, c'est plutôt « vous obtiendrez ce que vous demandez ». Un peu comme les fichiers en code HTML structurant

3. Prononcer [tek] (« tèque »), ou mieux [tex] comme le *ch* allemand dans *Bach, noch, Tuch*.

les pages Web qui s'affichent dans les navigateurs, un fichier TeX contient simplement le texte à mettre en forme, « au kilomètre », entrelardé de commandes et de déclarations. Dès lors, ce n'est bien sûr pas le fichier en question qui sera imprimé, mais par exemple le document PDF créé à partir du texte et de la suite d'instructions associées. Autrement dit, lorsqu'on travaille avec TeX, *on ne voit pas immédiatement le résultat de ce que l'on fait*.

Comme InDesign, la famille TeX permet elle aussi de réaliser tout type de documents, de la carte de visite à la partition pour orchestre symphonique⁴, en passant par les magazines ou les recueils de nouvelles. Il n'en faut pas moins admettre que la conception de maquettes complexes avec de tels programmes nécessite ou bien des capacités d'abstraction peu courantes, ou bien de travailler au préalable soit sur papier à l'ancienne, soit avec des outils graphiques... comme XPress ou Scribus!

Ce n'est donc pas sur le terrain des affiches, des revues ou des beaux livres qu'en matière de productivité les programmes de la famille TeX écrasent la concurrence. Si pourtant ils le font bel et bien, c'est évidemment lorsqu'il s'agit de produire d'autres documents, en l'occurrence des livres aux maquettes simples, tels romans ou essais.

ROMANS, ESSAIS : UNE EFFICACITÉ REDOUTABLE

« Une bonne typographie économise les dépenses de temps et les moyens. [...] Ce que nous faisons et la manière dont nous le faisons doit [...] correspondre à une nécessité évidente. » — Jan TSCHICHOLD (1902-1974)

Quand en effet la création d'une maquette se limite *grosso modo* à la définition du format de la page, des blancs tournants (les marges), des en-têtes et pieds de page, et au choix de la police d'écriture, de la force de corps et de l'interlignage, on peut se demander s'il est vraiment nécessaire d'avoir en permanence sous les yeux le « *what you get* ». D'autant que la plupart de ces paramètres sont en fait liés entre eux par des rapports mathématiques, ce qui réduit considérablement la part d'initiative personnelle : « Un livre réellement beau ne doit rien avoir de nouveau extérieurement, il doit tout simplement être parfait », écrivait encore Jan Tschichold.

Le premier point fort des programmes de la famille TeX, c'est la grande qualité typographique des documents produits obtenue sans intervention de l'utilisateur. Leur deuxième point fort est donc le gain de productivité impressionnant qu'ils permettent.

Ainsi, à partir d'un fichier ODT ou DOCX correctement préparé, et pour une maquette déjà définie, si l'on se contente de produire un ouvrage au « gris typographique⁵ » globalement satisfaisant — qui toutefois présentera ces défauts si fréquents dans l'édition littéraire française : superposition de mots identiques d'une ligne sur l'autre, mots coupés en bas de page, veuves et orphelines, notes s'étalant sur plusieurs pages en raison de la présence de l'appel

4. Avec ou sans le logiciel libre d'édition musicale LilyPond : fr.wikipedia.org/wiki/LilyPond.

5. « Dans un texte typographié : résultat de la combinaison du caractère utilisé, de ses attributs (sa force de corps, son œil, sa graisse, etc.), de l'interlignage et de la longueur de la justification [ligne pleine]. Chaque gris typographique traduit une atmosphère particulière qui convient ou ne convient pas à l'esprit du texte que l'on met en page. » (Atelier Perrousseaux)

correspondant dans les deux ou trois dernières lignes de la page —, quelques dizaines de secondes suffiront, avec l'aide d'un script shell ⁶, à produire un roman de trois cents pages. Aux défauts déjà évoqués, il faudra hélas ajouter une bonne trentaine de « lignes blanches ⁷ », mais le risque d'obtenir lézardes, rues ou cheminées est proche de zéro, tant les performances de la famille TeX sont élevées en matière de composition typographique. Pour peu que le texte à mettre en forme (la « copie ») ait été correctement stylé ⁸ par l'auteur ou les correcteurs, la structuration du document TeX (déclaration des titres de partie, de chapitre, de section...), mais aussi la composition des italiques et des gras ainsi que l'intégration des notes sont instantanées et ne requièrent aucune intervention, après l'exportation au format TeX du fichier produit par le logiciel de traitement de texte (Writer de LibreOffice, Word de Microsoft Office).

Avec la famille TeX, donc, moins d'une minute suffit à pondre un « livre moche à la française » (pour citer Olivier Bessard-Banquy) — et encore plutôt moins moche que la moyenne, puisque, outre la qualité supérieure du gris typographique, une opération très simple et quasiment instantanée ⁹ aura permis d'éviter les fins de ligne un peu mesquines comme « . À », « : à », « ? Ô », « ! Je » (proprement pitoyables en fin de belle page), pourtant monnaie courante chez les éditeurs à prix Goncourt.

Une fois la maquette définie, le metteur en page peut donc presque aussitôt se consacrer au traitement des divers problèmes qui relèvent de la typographie soignée. S'agissant d'un livre d'une justification moyenne de 58 à 60 signes, on peut envisager de progresser au rythme d'au moins 15 pages par heure, une bonne centaine par jour pour un résultat proche de la perfection, au moins égal à ce que proposent des éditeurs réputés en la matière comme Monsieur Toussaint-Louverture ou Allia (dont les poches laissent en fait souvent à désirer.)

Mais un peu d'histoire ne sera sans doute pas inutile pour comprendre d'où viennent TeX et ses descendants, et pourquoi ils sont si peu connus dans le secteur de l'édition littéraire et des sciences humaines, qui pourtant auraient cruellement besoin d'eux.

TEX

Donald Knuth ¹⁰ est né en 1938. Rien ne semblait prédestiner ce mathématicien et informaticien américain à concevoir l'un des tout meilleurs systèmes de composition typographique jamais mis au point. Rien sinon sans doute le père de ce dernier : professeur de comptabilité, mais surtout à la tête d'une petite imprimerie, Ervin Knuth, transmet très tôt à son fils le virus des livres et de la typographie.

C'est parce qu'il fut très mécontent de la piètre qualité des épreuves de la seconde édition du deuxième volume de sa série d'ouvrages *The Art of Computer Programming* qu'en 1977 Knuth décida de créer lui-même son propre système de composition de documents : TeX. Après la régression due à la photocomposition,

6. Programme informatique destiné à être exécuté par un interpréteur de commandes (par exemple, Bash pour les systèmes UNIX ou Windows PowerShell pour les systèmes de Microsoft). Voir fr.wikipedia.org/wiki/Interpréteur_de_commandes.

7. Dites aussi « lignes lavées » : lignes aux mots sensiblement trop espacés. Si en outre une ligne blanche se trouve encadrée par deux lignes à la composition serrée (mots moins espacés que dans la majorité des autres lignes), le défaut en devient criant.

8. Le stylage, dans un logiciel de traitement de texte comme de PAO, consiste à appliquer des styles de formatage, permettant de structurer le document (niveaux de titres, paragraphes...) et d'unifier la mise en forme.

9. Chercher-remplacer à l'aide d'une expression rationnelle. Voir l'article de Wikipédia : fr.wikipedia.org/wiki/Expression_rationnelle.

10. Prononcer « Quenoute », [kə'nu:θ].

il s'agissait d'obtenir une qualité au moins égale à celle de la composition au plomb traditionnelle, mais aussi de garantir la pérennité de ses travaux : « Depuis [ses] débuts [...], le projet de recherche de TeX dans lequel je m'étais embarqué répondait à deux grands objectifs. Le premier était celui de la qualité : nous ne voulions pas nous contenter de produire des documents qui fussent simplement jolis, nous voulions le meilleur résultat possible. [...] Le deuxième grand objectif était celui de l'archivage : créer des systèmes qui autant que possible ne seraient pas remis en cause par les changements à venir en matière de technologie d'impression. Lorsqu'apparaîtrait la prochaine génération de machines d'impression, je voulais pouvoir conserver le niveau de qualité déjà atteint, plutôt que de devoir résoudre à nouveau tous les problèmes qui s'étaient posés. Mon intention était de concevoir quelque chose que l'on pourrait toujours utiliser un siècle après. » — *Digital Typography*, p. 559 (traduction de Thomas Savary). Aujourd'hui, il est toujours possible, moyennant parfois quelques ajustements rapides, de produire des PDF à partir de fichiers TeX des années 1980. Quel autre programme de mise en page ou de traitement de texte pourrait en dire autant ?

En vue d'atteindre la qualité visée, il fallait permettre d'intervenir sur *tous* les aspects de la composition. À cette fin, Knuth conçut un langage informatique comprenant environ trois cents commandes (ou primitives), que le programme TeX se chargerait d'interpréter pour produire le document à imprimer. C'est ainsi qu'en 1978 une première version du programme vit le jour. Knuth avait notamment pris conscience que l'obtention d'un bon gris typographique nécessitait que le système de composition travaillât à l'échelle *non pas de la ligne, mais de l'alinéa*¹¹ : en fonction d'une hiérarchie de pénalités associées à des défauts de composition (définies par Knuth ou l'utilisateur), TeX essaie toutes les coupures de ligne et de mot possibles jusqu'à obtenir le nombre de points de pénalités minimal pour l'alinéa en cours de traitement. *Ce qui fut présenté par beaucoup comme une révolution au moment de la sortie d'InDesign en 1999, TeX le faisait donc déjà... en 1978.* Donald Knuth allait par ailleurs définir un ensemble d'environ six cents macrocommandes¹² appelé Plain TeX, destiné à la conception de ses propres ouvrages. Plain TeX permettait de simplifier l'usage de TeX, tout en offrant un exemple de programmation utilisant les primitives de ce dernier.

Enfin, en 1979, Knuth publia le langage Metafont, destiné à créer des polices d'écriture, avec lequel le scientifique élaborait lui-même la famille de polices Computer Modern, dont la famille dérivée Latin Modern est toujours utilisée aujourd'hui (en France, il m'a semblé reconnaître cette didone¹³ sur quelques couvertures de romans ou de beaux livres).

LATEX

Malgré ses possibilités presque illimitées, TeX, même avec Plain TeX, restait d'un accès difficile pour la plupart des gens. C'est la raison pour laquelle en 1983 Leslie Lamport, autre informaticien

11. Très souvent confondu avec le paragraphe depuis l'avènement de l'édition numérique. Si on s'en tient au cas général, un alinéa, dont la première ligne commence par un renforcement, est une portion de texte comprise entre deux renforcements de ce type. Un paragraphe, quant à lui, regroupe d'ordinaire plusieurs alinéas ; il est séparé du paragraphe qui suit ou qui précède par une ligne de blanc, un ou plusieurs astérisques (*), ou encore un astérisme (**).

12. Les macrocommandes sont en quelque sorte des miniprogrammes combinant une série de commandes élémentaires en vue d'exécuter une tâche précise.

13. Nommée d'après les graveurs Firmin Didot et Bodoni, cette famille de polices d'écriture créées à la fin du XVIII^e siècle se caractérise par leur verticalité ainsi que par des empattements rectilignes et des déliés d'une extrême finesse. Les caractères Firmin Didot furent utilisés en France dans la majeure partie des imprimés de 1810 aux années 1950.

américain, né en 1941, proposa un langage étendu de macrocommandes programmées elles-mêmes en TeX ou Plain TeX destiné à simplifier le recours au programme créé par Donald Knuth en déchargeant l'utilisateur de la plupart des aspects techniques de mise en forme. Plus besoin d'être un spécialiste de la typographie doublé d'un génie de l'informatique pour bénéficier de l'incroyable qualité permise par TeX. Le langage LaTeX (pour « L^ampo^rt TeX ») venait de voir le jour.

LaTeX tend à dissocier le fond de la forme. Il introduit la notion de classe de documents (lettre, article, rapport, transparent, livre...), associée chacune à sa propre mise en forme par défaut et à certaines commandes particulières. En contrepartie, les options de personnalisation des documents sont à la base limitées. Heureusement, par sa conception même, LaTeX peut être enrichi *ad libitum*. Non seulement l'utilisateur a tout loisir, directement dans le document à mettre en forme, de définir ses propres commandes en recourant au langage LaTeX, mais il lui est possible de faire appel à des extensions¹⁴ créées par d'autres personnes pour associer aux commandes LaTeX de base de nouvelles commandes obéissant à la même syntaxe, programmées elles aussi en TeX ou en Plain TeX. LaTeX présente ainsi l'avantage d'assurer une qualité typographique élevée tout en déchargeant l'utilisateur d'une grande partie des questions techniques, sans réduire pour autant les possibilités d'enrichissements et de personnalisation, pour les utilisateurs exigeants ayant des besoins particuliers.

Conçus par des mathématiciens, TeX et LaTeX sont particulièrement performants pour la composition des équations et des formules. C'est ce qui leur a valu leur succès dans l'édition des revues et livres scientifiques (mathématiques, physique, chimie, statistique...). Qui dit *travaux scientifiques* dit aussi *renvois, références, index, bibliographie, voire apparat critique*. C'est un des autres points forts de la famille TeX, qui n'est pas cette fois réservé aux sciences dures, mais intéresse aussi l'ensemble des sciences humaines.

Avec les extensions Reledmac et Reledpar de Maïeul Rouquette, LaTeX permet de réaliser des éditions critiques ou bilingues de grande qualité. Les références croisées (comme « voir note *x*, p. *y*. », alors qu'on n'a pour commencer aucune idée du numéro définitif de la note et de la page en question) n'ont jamais posé de problème à LaTeX. Si la création d'un index ne saurait se faire d'un claquement de doigts, l'extension Xindy de Roger Kehr et Joachim Schrod en simplifie grandement la mise en forme et garantit que les folios des renvois seront là encore toujours les bons, même en cas de changements de dernière minute dans le texte comme dans la maquette. Tous ceux qui par ailleurs se sont frottés à la création de bibliographies et à la citation des références correspondantes savent à quel point ces deux tâches se révèlent cauchemardesques. Mise au point par Philipp Lehman et Philip Kime, l'extension BibLaTeX, en association avec Biber, le logiciel de création de bibliographies conçu par le même Philip Kime avec François Charette, s'occupe justement de gestion des bibliographies et des renvois associés. Non seulement

14. Également appelées « modules » par certains (*packages* en anglais).

BibLaTeX et Biber réduisent les efforts au strict minimum (renseigner les informations comme « auteur », « titre », « éditeur », « date de parution », etc.), mais ce sont eux qui prennent en charge la mise en forme des entrées de la bibliographie comme des références : fini, les laborieux bricolages manuels approximatifs et les incohérences !

La multiplicité des classes et extensions de LaTeX permet en fait de répondre à pratiquement tous les besoins. Mais elle entraîne aussi parfois des problèmes agaçants : il arrive que certaines extensions interfèrent avec d'autres et provoquent des dysfonctionnements plus ou moins importants. En outre, la qualité de la documentation produite par leurs auteurs varie de l'excellent au trop succinct et mal rédigé. Avec son énorme lot de commandes supplémentaires et son manuel de plus de 600 pages (en anglais), la classe Memoir de Peter Wilson et Lars Madsen satisfait à la grande majorité des besoins d'un éditeur littéraire et limite considérablement le besoin d'appeler des extensions, donc le risque d'incompatibilités. Ce danger n'en est pas pour autant totalement écarté, dans la mesure où tôt ou tard il faudra ponctuellement recourir à telle ou telle extension.

CONTEXT

Cofondée par Hans Hagen, l'entreprise néerlandaise Pragma-ADE, spécialisée dans l'automatisation de la création de documents, a entrepris à son tour de mettre au point un système de composition unifié aux capacités étendues, à même de répondre à ses propres besoins tout en rendant l'usage de TeX accessible à l'ensemble de ses employés ou collaborateurs — à l'image de l'« écosystème » LaTeX, mais sur la base d'une approche fondamentalement différente et sans les inconvénients liés à la multiplication plus ou moins anarchique des extensions : lancé en 1990, ConTeXt se présente ainsi comme un nouveau langage de macrocommandes, programmées à l'origine, comme LaTeX, en TeX ou en Plain TeX. À partir de 1995, des personnes extérieures à Pragma-ADE commencèrent à utiliser ConTeXt, qui depuis n'a cessé d'évoluer.

Les commandes de ConTeXt sont très nombreuses et souvent plus puissantes que celles de LaTeX, avec abondance d'options. Texte, figures, tableaux... bien plus facilement que LaTeX, ConTeXt permet en outre de positionner sur la grille les différents éléments de la page, divisée en un grand nombre de zones. Traits de coupe, repères d'impression, gamme de couleurs et gamme de teintes, gestion de l'imposition... on comprend très vite que ConTeXt a été élaboré en fonction tant des besoins et attentes des imprimeurs que de ceux de l'édition.

ConTeXt compte cela dit bien moins d'utilisateurs à travers le monde que son aîné LaTeX, dans le domaine scientifique comme ailleurs. L'édition littéraire ? n'en parlons pas, où son potentiel pourtant est considérable. La documentation de ConTeXt dépend essentiellement d'un nombre très restreint de personnes, à commencer par Hans Hagen : ce qui existe est généralement clair et de qualité, mais des pans entiers restent à écrire. Nombreuses sont par exemple

les commandes à n'être toujours pas documentées — au moins en connaît-on la liste. Le wiki ConTeXtGarden ¹⁵ comprend quant à lui beaucoup d'articles non actualisés, avec de la documentation obsolète. Ici, nul n'est cependant à blâmer. Cet état de fait tient avant tout au nombre encore trop restreint des utilisateurs de ConTeXt, et alors que la plupart des problèmes relatifs à LaTeX ont déjà reçu réponse, et souvent à maintes reprises, sur les divers forums et sites techniques, il est beaucoup moins facile avec ConTeXt, en dehors de la liste de diffusion qui lui est consacrée, de trouver des réponses aux questions qu'on se pose.

15. fr.wikibooks.org/wiki/ConTeXt.

Les ressources en français, sans surprise, sont plus rares encore. Qui plus est, les options de ConTeXt sont actuellement loin de répondre aux nombreuses exigences spécifiques de notre typographie, ce qui empêche pour l'heure d'en recommander l'utilisation dans l'édition francophone. L'auteur de ces lignes entend toutefois bien contribuer à y remédier à moyen terme.

DES LOGICIELS LIBRES

À la différence de TeX et de LaTeX, ConTeXt a été publié sous licence GNU-GPL, la plus connue des licences rédigées par la Fondation pour le logiciel libre, créée en 1985 et présidée jusqu'en 2019 par Richard Stallman (Free Software Foundation).

Pour mémoire, voici les quatre règles fondatrices du logiciel libre :

- liberté d'exécuter le programme pour tous les usages (liberté 0);
- liberté d'étudier le fonctionnement du programme, et de l'adapter à ses besoins (liberté 1); pour cela, l'accès au code source est une condition requise;
- liberté d'en redistribuer des copies, donc d'aider son prochain (liberté 2);
- liberté d'améliorer le programme et de publier des améliorations, pour en faire profiter toute la communauté (liberté 3); à cette fin, l'accès au code source est également une condition requise.

À la lecture des pages qui précèdent, on pouvait en effet se demander comment le projet lancé par Donald Knuth a pu se voir ainsi prolongé et enrichi par tant de contributeurs différents, souvent sans liens entre eux. C'est que, sans être tout à fait compatibles avec la licence GNU-GPL, TeX et LaTeX n'en sont pas moins des logiciels libres, dans la mesure où aucune restriction n'est appliquée à leur usage, que le code source de ces programmes est public et qu'ils sont librement copiables et redistribuables. Sont également permises les modifications, mais cette fois avec l'interdiction de désigner par le nom de TeX les versions modifiées (différence importante avec la licence de la Free Software Foundation).

Comme Linux, Firefox, LibreOffice, Calibre, GIMP, Krita, Audacity, Clementine ou encore le lecteur multimédia VLC, c'est bien parce que TeX et LaTeX sont des logiciels au code source ouvert (« *open source* »), librement utilisables et redistribuables (donc disponibles

gratuitement) qu’au fil des ans ils ont connu un tel nombre d’enrichissements — jusqu’à, pour TeX, la réécriture complète, tout en conservant les primitives d’origine et Plain TeX.

SUCESSEURS DE TEX

Diverses versions modifiées de TeX ont donc pu voir le jour. Citons pdfTeX (2000, perfectionnant l’algorithme de coupure de ligne de Knuth tout en apportant des améliorations très appréciables en matière de microtypographie¹⁶ et en permettant de créer des PDF, format de fichier qui n’existait pas lorsque TeX fut mis au point), XeTeX¹⁷ (2004, qui permet aux utilisateurs d’accéder aux polices du système et travaille en UTF-8¹⁸), ou encore LuaTeX (2007, basé sur un pdfTeX complètement réécrit, avec les mêmes apports que XeTeX et bien d’autres). C’est d’ailleurs désormais sur LuaTeX, également sous licence GNU-GPL, que repose ConTeXt. En avril 2019, Hans Hagen a annoncé la mise en chantier de LuaMetaTeX (LMTX) ou LuaTeX 2, nouvelle version de LuaTeX destinée spécifiquement à ConTeXt, qui ne sera plus compatible avec l’écosystème LaTeX. LuaTeX continuera cependant d’être maintenu et pourrait même finir par bénéficier de certaines nouvelles fonctions mises au point pour LMTX¹⁹.

Pour des raisons historiques, TeX, conçu avant l’explosion de l’informatique personnelle, gérait de manière autonome les polices d’écriture à sa disposition, sans pouvoir accéder à celles installées sur le système de l’utilisateur. TeX a en outre été publié bien avant l’apparition, au début des années 1990, du codage UTF-8. Ainsi XeTeX puis LuaTeX ont-ils été élaborés notamment pour simplifier l’utilisation des polices (en rendant enfin accessibles celles installées sur le système d’exploitation), mais aussi pour permettre de travailler avec des textes codés directement en UTF-8.

Par ailleurs, même avec les macros Plain TeX ou LaTeX, les possibilités en matière de programmation restaient limitées. Or, la création de certains graphiques ou tableaux requiert de véritables outils de programmation qui dépassent les questions en rapport avec la typographie. Utilisable en parallèle avec les commandes LaTeX ou ConTeXt à l’intérieur même des documents à mettre en forme, le langage de programmation Lua intégré à LuaTeX permet d’interagir avec le compositeur. Voilà qui n’est pas seulement utile pour les publications scientifiques, mais qui entre autres a permis à Raphaël Pinson de programmer une extension LaTeX signalant la superposition sur plusieurs lignes successives de mots identiques ou homographes, ce qui est d’une grande aide pour améliorer la composition de tout type de document, y compris et surtout les textes littéraires²⁰.

INDESIGN OU LATEX ?

Pour la conception d’ouvrages nécessitant une maquette élaborée, il est indéniable que des programmes comme InDesign, QuarkXPress,

16. Voir pour une justification étroite ce test comparatif réalisé en 2010 entre Microsoft Word, Adobe InDesign CS4 et pdfTeX, qui se démarque nettement) : rtznet.nl/zink/comparison.pdf.

17. Prononcer [zi:tek] (« zitéque ») ou [zi:tex].

18. Codage de caractères permettant de gérer le plus simplement possible des textes et leurs traductions dans tous les systèmes d’écriture et tous les alphabets du monde.

19. Voir « ConTeXt LMTX », *TUGboat*, vol. 40 (2019), numéro 1, p. 34-37, tug.org/TUGboat/tb40-1/tb124hagen-lmtx.pdf.

20. Baptisée Impnattypo, elle permet en outre de détecter les débuts de lézarde, rue ou cheminée, tout en contribuant à diminuer le nombre de « lignes à voleur » (lignes creuses de moins de cinq ou six caractères).

Affinity Publisher ou bien Scribus sont d'un abord plus facile et offrent une meilleure productivité — encore que ConTeXt, conçu notamment pour simplifier l'automatisation de la mise en pages de documents complexes, soit peut-être en mesure ici de rivaliser (l'auteur de ces lignes ne maîtrise pas encore suffisamment ConTeXt pour se prononcer). S'agissant en revanche d'un recueil de nouvelles ou d'un roman, d'un essai (agrémenté ou non d'illustrations, de schémas, de graphiques et de tableaux), il est indéniable que les logiciels de la famille TeX l'emportent haut la main concrètement.

Il est vrai qu'InDesign permet l'utilisation de scripts très puissants qui rendent possible la mise en pages automatique à la chaîne. Les graphistes semblent pourtant bien rares à les utiliser. Faut-il s'en étonner, dans la mesure où des concepts mêmes d'interface graphique et de WYSIWYG découlent des méthodes de travail aux antipodes de la programmation impérative ou déclarative ? Non que les deux approches soient finalement incompatibles, comme le montre l'exemple d'InDesign, mais la capacité d'un individu à les associer suppose une certaine approche de l'informatique qui à l'évidence n'est pas celle de la majorité des graphistes, sans qu'ici bien sûr il faille jeter la pierre à Adobe. InDesign n'en présente pas moins un sérieux problème en matière de programmation dans la mesure où les évolutions du logiciel frappent souvent d'obsolescence tout ou partie de nombreux scripts antérieurs, ce qui contraint à un travail de réécriture parfois considérable (sujet de mécontentement récurrent chez les utilisateurs avancés de ce programme).

Tout n'est assurément pas parfait à cet égard dans le monde des programmes et langages de la famille TeX : le choix de changer de moteur (pdfTeX, XeTeX, LuaTeX, LMTX) ou même les évolutions d'une classe comme d'une extension sont susceptibles d'entraîner des différences, souvent légères, mais qui en cascade peuvent vite obliger à reprendre tout ou partie du travail pour un ouvrage donné (surtout dans le cas de maquettes complexes calibrées au millimètre). Pour les classes et les extensions, rien n'empêche cela dit de continuer à utiliser des versions précédentes sans renoncer aux nouveautés par ailleurs : c'est là tout l'avantage de la modularité de LaTeX et des logiciels libres, dont les anciennes versions restent longtemps disponibles — au pire, à l'utilisateur d'effectuer des copies de sauvegarde des fichiers de classe et d'extension qu'il utilise, pour au besoin les inclure dans ses dossiers de travail. Les scripts permettant l'automatisation peuvent quant à eux rester inchangés, car ils sont écrits dans des langages indépendants de la famille TeX et de ses évolutions : Perl, Ruby, Python, scripts shell... C'est pourquoi, dans le domaine de l'automatisation, les programmes de la famille TeX paraissent une solution non seulement au moins aussi efficace qu'InDesign, mais plus naturelle pour l'utilisateur dans le cadre d'un travail en mode texte plutôt qu'en WYSIWYG, et assurément beaucoup plus sûre.

Mais *quid* de l'opérateur prépresse, du maquetiste ? Ce dernier, s'il est amené à créer toutes sortes de documents, a-t-il quant à lui intérêt à investir le temps nécessaire à l'apprentissage de la mise

en page et de la composition avec les logiciels de la famille TeX ? Probablement pas, car il faut reconnaître que cet apprentissage ne sera pas *a priori* des plus simples pour quelqu'un n'ayant jamais travaillé qu'à travers des interfaces graphiques. Si toutefois ce maquetiste a souvent l'occasion de mettre en pages des livres comprenant essentiellement, voire uniquement du texte, il serait bien inspiré au moins d'apprendre ou de réviser les bases du métier... À quoi bon en effet disposer d'un logiciel performant comme InDesign, il faut le reconnaître, si c'est pour ignorer les différentes méthodes d'empagement ou ne pas savoir, par exemple, qu'on évite de couper le dernier mot d'une page de même, autant que possible, que les noms propres ; que la dernière ligne d'un alinéa doit comporter au moins cinq ou six caractères ; que cette dernière ligne devrait sinon être suffisamment creuse²¹ (d'au moins un cadratin²²), ou alors pleine (pis-aller pour certains) ; qu'après une lettrine les lettres suivantes du premier mot, voire de tout le groupe de mots qui suit se composent traditionnellement en petites capitales ; que, dans un dialogue, l'espace qui suit le tiret doit avoir une chasse fixe pour assurer l'alignement vertical des premières lettres de chaque réplique ; qu'on évitera autant que possible de laisser des mots identiques superposés d'une ligne sur l'autre, etc. ?

Combien ai-je pu lire de romans défigurés par d'innombrables problèmes de composition ou une mise en page conçue en dépit du bon sens, tout produits par InDesign ou XPress qu'ils aient été ! On peut certes aussi créer des horreurs avec LaTeX, mais c'est plus difficile, car il faut le vouloir ! Plus sérieusement, disons que LaTeX n'a jamais protégé des empagements catastrophiques, dès lors que l'on commence à faire joujou avec l'extension Geometry, mais que la qualité des systèmes de composition associés, qu'il s'agisse de TeX, de pdfTeX, de XeTeX ou de LuaTeX, est beaucoup plus rarement prise en défaut, même entre des mains novices, que dans ce que donnent à voir un InDesign ou un XPress insuffisamment maîtrisés.

Mettre en pages et composer un roman, une monographie, un essai, une pièce de théâtre ou bien réaliser un beau livre, un magazine, une affiche... ce ne sont au fond pas les mêmes métiers : la part d'invention et de créativité est faible, voire nulle dans le premier cas, alors qu'elle est bien évidemment importante, voire essentielle, dans le second. Il est finalement logique de ne pas recourir aux mêmes outils selon la nature des textes auxquels on travaille.

Si son métier consiste à mettre en pages essentiellement, voire uniquement des ouvrages aux maquettes simples, alors il n'existe probablement pas d'outils plus adaptés, performants, rapides et rentables que ceux de la famille TeX — en dehors peut-être de systèmes de publication extrêmement coûteux comme Arbortext Advanced Print Publisher. « Accessoirement », les programmes de la famille TeX, eux, sont gratuits.

21. Par opposition aux lignes pleines, une ligne est dite creuse lorsqu'elle n'occupe pas toute la justification (largeur du bloc de texte).

22. Unité de mesure de longueur égale à la force de corps de la police utilisée, équivalant normalement à la « chasse » (largeur) de deux zéros en haut de casse (00).

PÉRENNITÉ ET MAÎTRISE DES DONNÉES

Parue en 2012, la dernière version autonome d'InDesign (CS6) coûtait aux alentours de 900 EUR. Depuis la version suivante, publiée en 2014, InDesign ainsi que les autres composants de la Creative Suite d'Adobe ne sont plus disponibles que sur abonnement au « Creative Cloud ». Fini, les licences de logiciels sans limitation de durée ! Adobe s'inscrit donc dans la tendance suivie par les gros éditeurs de logiciels propriétaires : l'utilisateur n'achète plus un programme, mais s'abonne à un service — moins que jamais, on est maître de « ses » programmes et de sa machine. Pour InDesign, il en coûte actuellement 23,99 EUR par mois, soit près de 290 EUR par an, plus de 1 400 EUR sur cinq ans... Pas sûr qu'il s'agisse là d'une bonne affaire, sinon bien sûr pour ceux qui achetaient systématiquement chaque nouvelle version du logiciel — pour de simples romans, toutefois, était-ce là encore vraiment nécessaire ?

Le « Cloud » présente d'indéniables avantages pratiques, mais n'oublions pas la réalité : « *There is no Cloud, just other people's computers* » (« Le Cloud n'existe pas : ce sont juste les ordinateurs de quelqu'un d'autre » — Free Software Foundation). Quel est l'intérêt pour un éditeur de livres de confier à une entreprise tierce l'accès partiel à ses propres ordinateurs ou aux ordinateurs de ses prestataires, voire, dans le cas de certains « services », aux données qui lui appartiennent ?

Autre sujet d'importance que soulevait Donald Knuth dès la fin des années 1970, celui de la pérennité des données. En proposant mes services à des éditeurs, j'ai obtenu plus d'une fois ce genre de réponse lorsque je précisais que je n'utilisais pas le programme d'Adobe : « Ce programme [LuaTeX] est sans doute très bien, mais nous utilisons InDesign, et nous tenons à pouvoir conserver les fichiers originaux pour d'éventuelles rééditions ou le passage en poche. »

Outre qu'il paraît bien plus judicieux, dans le deuxième cas, de repartir de la copie (le tapuscrit corrigé et mis en forme) plutôt que du livre maqueté dans un autre format, c'est bien l'utilisation de logiciels propriétaires qui peut rendre épineuse la question des nouvelles éditions ! Au fur et à mesure des nouvelles versions de ces logiciels, leur format de fichier évolue et la rétrocompatibilité avec les formats antérieurs n'est pas assurée indéfiniment. Un confrère utilisant QuarkXPress m'a avoué que pour la deuxième édition d'un ouvrage qu'il avait composé et mis en pages, il lui avait fallu douze ans plus tard tout reprendre à zéro avec la dernière version en date du programme de Quark ! Dans InDesign, le passage par le format IDML permet au moins de récupérer le fichier. Ce sont les scripts, on l'a vu, qui pourront poser le plus de problèmes. Trente ans après, les fichiers LaTeX, au moins pour les documents peu complexes, demandent peu d'ajustements, voire aucun, pour composer des PDF destinés à l'impression. Qui dit mieux ? Et même si au pire il fallait tout reprendre, au moins les choses iraient-elles vite, du fait de la productivité de la famille TeX pour l'édition littéraire.

À l'heure de la littérature jetable, cette question de la pérennité des données a peut-être peu d'importance. Il me semble toutefois qu'elle en a encore beaucoup pour les maisons d'édition qui ambitionnent de faire vivre leur fonds de catalogue sans dépendre du bon vouloir d'un éditeur de logiciels pour qui plus que jamais, à travers ses formules d'abonnement, le client n'est qu'une vache à lait.

Si Adobe InDesign évolue, ce n'est en outre pas forcément dans le bon sens, à en juger par les déclarations d'un des meilleurs spécialistes de ce logiciel, Branislav Milić, dans un entretien publié en 2014 par le site *MacGeneration.com*²³. Entre-temps, les choses ne se sont guère arrangées, à l'en croire. Sur son site en 2019, Milić dépeint Adobe comme « une boîte californienne de marketing avec des développeurs en Inde » : ingénieurs et programmeurs originaux d'InDesign ont en effet été remerciés dès 2014. Milić déplore l'attitude hautaine d'Adobe et de ses nouveaux développeurs, refusant de prendre en compte les demandes des professionnels au profit de nouveautés marketing sans grand intérêt²⁴. Voilà qui paraît à l'exact opposé des évolutions de ConTeXt et de LuaTeX, dont les concepteurs sont restés très à l'écoute des utilisateurs.

23. « Adobe InDesign : 15 ans, l'âge bête », interview menée par Mickaël Bazoge, *MacGeneration*, www.macg.co/logiciels/2014/08/adobe-indesign-15-ans-lage-bete-83891.

24. « Adobe InDesign a 20 ans », branislavmili.com/indesign/index.html.

LIBERTÉ ET QUALITÉ

Pourquoi choisir les logiciels libres chaque fois que c'est possible ? Tout simplement pour rester maître de ses outils comme de son travail. Avec les programmes de la famille TeX, on est libre de choisir son système d'exploitation, quand InDesign vous impose Mac OS X ou Windows ; libre de choisir son compositeur, TeX, pdfTeX, XeTeX ou LuaTeX ; libre de choisir son langage, TeX et Plain TeX, LaTeX ou ConTeXt, voire Lua ; libre de choisir son éditeur, Kile, Texmaker, TeXworks, TeXstudio, TeXnicCenter, LaTeXila, ShareLaTeX ou bien d'autres comme Vim, GNU Emacs, et même un programme aussi basique que le Bloc-notes de Windows... Certes, les logiciels libres ne sont pas toujours les plus performants. Par chance, c'est pourtant ici bien le cas avec la famille TeX, s'agissant de répondre aux besoins de l'édition littéraire ou des sciences humaines.

Si un livre médiocre a tout à gagner d'une mise en pages réussie, le chef-d'œuvre à l'inverse résistera aux injures du travail bâclé. Il n'en demeure pas moins qu'une mise en pages ratée, voire seulement médiocre atteste un mépris du livre comme du lecteur... Qu'il achète, qu'il consomme ! Serait-ce donc tout ce qui compte ?

Bien que ne parlant pas le polonais, c'est toujours avec plaisir que dans la bibliothèque d'une amie j'ouvre au hasard les livres imprimés en cette langue. Est-ce par chance si parmi ceux que j'examine le moindre des ouvrages produits ces vingt-cinq dernières années est du niveau de ce qui en France se fait de mieux en matière de typographie dans l'édition littéraire ?

Gros, petits, moyens, trop nombreux hélas sont les éditeurs qui semblent ne pas accorder la moindre importance à la mise en pages des livres qu'ils publient. Quitte à passer pour un décliniste aigri

et réactionnaire, oserai-je rappeler que la France fut durant trois siècles un des hauts lieux de la typographie ? Comment tant de maisons se targuant de publier des livres justifieront-elles leur incurie ? Et si au lieu de jeter leur argent par les fenêtres ou bien de se tourner comme c'est de plus en plus souvent le cas vers des prestataires meilleur marché et corvéables à merci utilisant InDesign, au Maroc ou encore à Madagascar, les éditeurs commençaient par privilégier enfin les solutions les plus efficaces, les plus pérennes et les moins coûteuses ?

Je ne crois pas être le seul lecteur fatigué des « livres moches à la française »... Sur la Toile, qui n'eut longtemps à offrir aux amateurs de belle composition que des visions cauchemardesques, on a vu renaître un intérêt pour la typographie : le couple HTML-CSS progresse, il est enfin possible de composer de manière acceptable ; les sites sur le sujet se multiplient (peu en français, à ma connaissance) ; les étudiants semblent de plus en plus nombreux à chercher de meilleures solutions que Microsoft Word ou LibreOffice Writer pour mettre en pages leurs mémoires sans devoir pour autant se ruiner. De nouvelles maisons d'édition se créent qui n'ont rien à gagner à l'usage d'InDesign. À l'heure de la multiplication des supports de l'écrit et des formats numériques pour le texte, les langages de mise en page et de composition ainsi que les formats ouverts sont plus d'actualité que jamais.

PROLONGEMENTS

L'auteur

« Je ferais mieux si j'apprenais à me servir d'un logiciel de PAO... » Voilà ce que longtemps je me suis répété à force de corriger des épreuves mal mises en pages parfois, présentant de multiples problèmes de composition toujours. J'avais déjà conçu plusieurs sites en HTML et CSS. Ainsi mon choix s'est-il finalement porté sur LaTeX, associé au système de composition LuaTeX. Depuis l'automne 2015, je mets en pages et compose romans et nouvelles, essais, pièces de théâtre... En 2019, j'ai commencé à proposer mes services de création de livres numériques (EPUB, KEPUB, MOBI).

On trouvera sur mon site quelques exemples de mes réalisations dans la rubrique « Édition papier. — Exemples » : www.compo85.fr.

Thomas Savary, 1 le Grand-Plessis, 85340 L'Île-d'Olonne
Tél. 06 22 82 61 34 – Mél. compo85@correctionpro.fr
CV : www.compo85.fr/docs/CV_PAO_Thomas_Savary.pdf

Réalisations par d'autres utilisateurs de TeX & C^{ie}

Parmi les exemples suivants, on découvrira des maquettes complexes, qui démontrent que tout est possible avec la famille TeX, même si, pour la majorité des graphistes, des programmes WYSIWYG comme InDesign, XPress ou Scribus paraissent plus indiqués.

- tex.stackexchange.com/questions/1319/showcase-of-beautiful-typography-done-in-tex-friends
- www.tug.org/texshowcase/; dans la catégorie « General Type-setting », voir plus particulièrement *The Book of Tea*²⁵ ainsi qu'*Exiles from a Future Time*²⁶ ;
- tsengbooks.com/, très beaux extraits des livres de cet éditeur utilisant les logiciels de la famille TeX.

25. www.tug.org/texshowcase/partofTheBook-ofTea.pdf.

26. www.tug.org/texshowcase/6553-sample.pdf.

Articles de Wikipédia

- WIKIPÉDIA. *TeX*. 2020. URL : <http://fr.wikipedia.org/w/index.php?title=TeX> (visité le 2020-02-29).
- WIKIPÉDIA. *LaTeX*. 2020. URL : <http://fr.wikipedia.org/w/index.php?title=LaTeX> (visité le 2020-02-29).
- WIKIPÉDIA. *ConTeXt*. 2019. URL : <http://fr.wikipedia.org/w/index.php?title=ConTeXt> (visité le 2020-02-29).
- WIKIPÉDIA. *PdfTeX*. 2019. URL : <http://fr.wikipedia.org/w/index.php?title=PdfTeX> (visité le 2020-02-29).
- WIKIPÉDIA. *XeTeX*. 2019. URL : <http://fr.wikipedia.org/w/index.php?title=XeTeX> (visité le 2020-02-29).
- WIKIPÉDIA. *LuaTeX*. 2020. URL : <http://fr.wikipedia.org/w/index.php?title=LuaTeX> (visité le 2020-02-29).

Documentation en ligne

- WIKILIVRES. *LaTeX*. 2016. URL : <https://fr.wikibooks.org/w/index.php?title=LaTeX> (visité le 2020-02-29).
- WIKILIVRES. *ConTeXt*. 2017. URL : <https://fr.wikibooks.org/w/index.php?title=ConTeXt> (visité le 2020-02-29).
- Groupe francophone des utilisateurs de TeX, LaTeX et logiciels compagnons. URL : <https://www.gutenberg.eu.org/> (visité le 2020-02-29).
- GRUPE FRANCOPHONE DES UTILISATEURS DE TEX. *Cahiers GUTenberg*. 1988–2012. URL : <http://cahiers.gutenberg.eu.org/> (visité le 2020-02-29).
- MASSON (Bertrand). *Les Fiches à Bébert. Où l'on parle de TeX, LaTeX, ConTeXt & co*. URL : <http://www.lesfichesabebert.fr/> (visité le 2020-02-29).
- Comprehensive TeX Archive Network*. URL : <https://ctan.org/> (visité le 2020-02-29).

Livres et manuels PDF

LaTeX

- CHEVALIER (Céline). *LaTeX pour l' impatient*. 4^e édition, revue, corrigée et augmentée. « MiniMax ». Paris : H&K, 2016. 192 pages. ISBN 9782351413272.
- DESGRAUPES (Bernard). *LaTeX. Apprentissage, guide et référence*. 3^e édition, revue et corrigée. « Informatique ». Louvain-la-Neuve : De Boeck supérieur, 2019. 816 pages. ISBN 9782807323629.

ROUQUETTE (Maïeul). *(Xe)LaTeX appliqué aux sciences humaines*. Tampere : Atramenta, 2012. 270 pages. ISBN 9789522730732. URL : <https://halshs.archives-ouvertes.fr/halshs-00924546/document> (visité le 2020-02-29).

WILSON (Peter) et MADSEN (Lars). *The Memoir Class for Configurable Typesetting User Guide*. 8^e édition. Normandy Park : The Herries Press, 2009. 615 pages. URL : <http://mirrors.ctan.org/macros/latex/contrib/memoir/memman.pdf> (visité le 2020-02-29).

ConTeXt

OTTEN (Ton). *ConTeXt, an Excursion*. Hasselt : Pragma ADE, 2017. 170 pages. URL : <http://pmrb.free.fr/contextref.pdf> (visité le 2020-02-29).

HAGEN (Hans) et HOEKWATER (Taco). *ConTeXt Reference Manual*. Hasselt : Pragma ADE, 2013. 376 pages. URL : <http://pmrb.free.fr/contextref.pdf> (visité le 2020-02-29).

HAGEN (Hans). *Typographic Programming*. Hasselt : Pragma ADE, 2004. 93 pages. URL : <http://www.pragma-ade.com/general/manuals/style> (visité le 2020-02-29).

Moteurs TeX

ABRAHAMS (Paul), HARGREAVES (Kathryn) et BERRY (Karl). *TeX pour l'impatient*. Traduit par Marc CHAUDEMANCHE. Boston : Free Software Foundation, 2004. 407 pages. URL : <http://mirror.ctan.org/info/impatient/fr/fbook.pdf> (visité le 2020-02-29).

EIJKHOUT (Victor). *TeX by topics. A TeXnician's Reference*. Victor Eijkhout, 2019. 325 pages. URL : <https://bitbucket.org/VictorEijkhout/tex-by-topic/src/default/> (visité le 2020-02-29).

KNUTH (Donald Ervin). *Le TeXBook. Composition informatique*. Traduit par Jean-Côme CHARPENTIER. Paris : Vuibert, 2003. 556 pages. ISBN 9782711748198.

TELLECHEA (Christian). *Apprendre à programmer en TeX*. Christian Tellechea, 2014. 580 pages. URL : <http://www.lulu.com/shop/christian-tellechea/apprendre-%C3%A0-programmer-en-tex/paperback/product-21816783.html> (visité le 2020-02-29).

ROBERTSON (Will), HOSNY (Khaled) et KARL (Berry). *The XeTeX Reference Guide*. Heidelberg : Comprehensive TeX Archive Network, 2019. 25 pages. URL : <http://mirrors.ctan.org/info/xetexref/xetex-reference.pdf> (visité le 2020-02-29).

CONCEPTEURS DE LUALATEX. *LuaTeX Reference Manual*. Hasselt : Pragma ADE, 2019. 294 pages. URL : <http://www.pragma-ade.com/general/manuals/luatex> (visité le 2020-02-29).

Aide en ligne

TeXnique.fr. URL : <http://texnique.fr> (visité le 2020-02-29).

TeX-LaTeX Stack Exchange. Site anglophone. URL : <https://tex.stackexchange.com> (visité le 2020-02-29).