

# La

# LETTRE

## Numéro 50 – juin 2023

Activité récente de l'association .....	3
Exposés mensuels sur $\text{\LaTeX}$ et autres logiciels .....	6
De nouveaux packages sur le CTAN .....	7
Un site dédié aux exemples METAPOST .....	22
Composer l'arbre de Huffman avec METAPOST et METAOBJ ...	26
Dans le labyrinthe infernal des environnements d'amsmath ..	42
La fonte de ce numéro : Noto .....	47
Publicité .....	52
En route vers Noto .....	53
Brèves .....	74
Exposé à venir : « Introduction à LuaLaTeX » .....	79
Rébus .....	80
Acronymes .....	80
Adhésion à l'association .....	81

Chers adhérents,

cette *Lettre* contient de longs articles, consacrés à METAFONT d'une part et aux fontes Noto de l'autre. Nous avons choisi de composer ce numéro avec cette famille de fontes car il nous a semblé utile de mettre ce vaste travail en valeur : de l'Adlam au Zanabazar Square, les fontes Noto permettent de composer plus de 1000 langues dans plus de 150 systèmes d'écriture différents ! Nous vous renvoyons à l'article de Steven Matteson en page 53 pour plus d'informations. Mais le choix de cette famille de fontes ne signifie pas que nous cautionnons les agissements de la société Google, qui est à l'origine de ce projet. Nous ne nous étendons pas plus sur ce sujet et renvoyons les lecteurs à l'article de Wikipédia<sup>1</sup> synthétisant les critiques à l'encontre de cette société.

Cette *Lettre* fait la part belle à METAPOST. Le monde de  $\text{\TeX}$  regorge d'outils pour dessiner ou faire des représentations graphiques. METAPOST, héritier

1. [https://fr.wikipedia.org/wiki/Critiques\\_de\\_Google](https://fr.wikipedia.org/wiki/Critiques_de_Google)

**Avez-vous pensé à régler votre cotisation ?**

Si vous avez oublié, ce n'est ni trop tard ni difficile :  
<https://www.gutenberg-asso.fr/?Adherer-en-ligne>

# GUTenberg

direct de METAFONT (apparu en même temps que T<sub>E</sub>X), fait partie des plus anciens, puisque sa version 1.0 date de 1994<sup>2</sup>. Pour information, parmi les figures de proue du graphisme dans le monde de T<sub>E</sub>X, P<sub>S</sub>Tricks est né lui en 1993, et TikZ en 2005. Ces deux derniers langages (ou plutôt extensions du langage T<sub>E</sub>X) proposent de bien plus nombreux outils que METAPOST. Cela est sans doute dû au fait que ces *extensions* de T<sub>E</sub>X permettent de générer leurs figures directement lors de la compilation d'un document L<sup>A</sup>T<sub>E</sub>X (ou autre). Tandis que, antérieurement à l'avènement de LuaT<sub>E</sub>X<sup>3</sup>, l'inclusion des figures METAPOST n'était pas immédiate puisqu'il fallait les générer à part et les inclure dans le document L<sup>A</sup>T<sub>E</sub>X avec un `\includegraphics`. Pour la gestion des fontes notamment, ça n'était pas le plus facile — même si, une fois compris le mécanisme, cela ne posait pas de problème majeur. Les choses ont changé avec LuaT<sub>E</sub>X puisque désormais METAPOST fait partie intégrante du moteur : l'intégration de figures réalisées avec METAPOST se fait très simplement avec, pour L<sup>A</sup>T<sub>E</sub>X, le package `luamplib`<sup>4</sup>. Même si METAPOST n'est pas le système le plus utilisé par la communauté, il reste très puissant et très bien conçu<sup>5</sup>. Vous pourrez d'ailleurs (re)découvrir une extension de METAPOST nommée METAOBJ, elle aussi très puissante, produite par un ancien membre du CA de l'association et contributeur important du monde de T<sub>E</sub>X : Denis Roegel.

Vous ne trouverez pas dans cette *Lettre* de compte rendu de réunion : le conseil d'administration ne s'est pas réuni depuis la parution du précédent numéro. Non que nous soyons restés inactifs : les membres du bureau sont en contact quotidien via la messagerie Signal et Jean-Michel Hufflen travaille au prochain numéro des *Cahiers*. Vous lirez en page suivante les différentes activités qui furent les nôtres ces derniers mois. Certains auront remarqué que nous avons annoncé la sortie de la présente *Lettre* il y a quelques semaines ; nos autres tâches en ont repoussé la parution.

L'association est au service de la communauté des utilisateurs francophones de T<sub>E</sub>X, dont certains participeront à la conférence TUG 2023 qui a lieu à Bonn (Allemagne) du 14 au 16 juillet prochains (cf. page ci-contre). C'est avec plaisir que nous constatons la vitalité T<sub>E</sub>Xoïdale francophone, notamment lors de la publication de nouveaux packages sur le CTAN. Comme le montre notre *veille T<sub>E</sub>Xnologique* (voir page 7), dans laquelle nous les recensons, livraison après livraison, il est rare que s'écoule un mois sans qu'un francophone en publie un<sup>6, 7</sup>.

Auteurs de packages, comme le fait Maxime Chupin dans son article sur l'arbre de Huffman en page 26, décrivez la réalisation de vos packages ! Étape par étape ! De l'idée au package fonctionnel ! Nous publierons volontiers vos travaux, et nul doute qu'ils susciteront des vocations !



Patrick Bideault & Maxime Chupin

2. Mais son auteur, J. D. Hobby, publie une présentation dans le TUGBoat en 1989.

3. Taco Hoekwater a séparé le programme METAPOST en deux parties en développant la bibliothèque `mplib` autour de 2007, qui a été intégrée à LuaT<sub>E</sub>X.

4. Voir l'article *LuaLaTeX et MetaPost avec luamplib, une introduction* dans les *Cahiers* 58.

5. METAPOST reprend presque intégralement le code de METAFONT !

6. Parmi les nouveautés, ces packages sont faciles à reconnaître : le logo de l'organisation internationale de la francophonie figure à gauche de leur nom.

7. Depuis que, dans la *Lettre* 42, nous recensons les nouveaux packages — nous mettons un point d'honneur à les recenser avec exhaustivité — la moyenne mensuelle de parution de nouveaux packages écrits par des francophones s'est établie à 1,68 (soient 54 packages en 32 mois... et celui-ci n'est pas fini !). Ce chiffre nous semble tout à fait honorable et l'association s'enorgueillit de participer pour une part modeste, mais déterminée, à l'activité autour de nos

## ACTIVITÉ RÉCENTE DE L'ASSOCIATION

Cahiers .....	3
Un site consacré à METAPOST .....	3
Conférence TUG 2023 à Bonn .....	3
Fontes anciennes .....	4
CTAN .....	4
Sondage .....	4
DOI .....	5
Pourriels .....	5
Trésorerie .....	5
Dolibarr .....	5
Exposés .....	5

### Cahiers

Il va bientôt sortir... le *Cahier* 59!

Le prochain numéro va décrire quelques utilisations peu courantes de  $\LaTeX$  : dans le cadre d'ouvrages de droit (par Flora Vern) et de philologie (par Maieul Rouquette), pour des éditions de textes classiques (par Thomas Savary). Également au programme : Laurent Bloch, qui nous présente une version révisée et augmentée de « Efficacité comparée de  $\LaTeX$  et MS Word », lisible sur le web. Ainsi que d'autres rubriques.

La souscription va bientôt être ouverte, mais nous communiquerons spécifiquement sur le sujet le moment venu.

### Un site consacré à METAPOST

Depuis février dernier, le serveur de l'association recèle un site consacré à METAPOST. Maxime Chupin y consacre un article en page 22.

<https://metapost.gutenberg-asso.fr>

### Conférence TUG 2023 à Bonn

La conférence annuelle du TUG ayant lieu en Europe<sup>8</sup>, l'association a décidé d'attribuer deux bourses à des membres désirant s'y rendre, charge à eux de rendre compte de la conférence ou d'y présenter leur travail lors d'une conférence. Or les deux bourses proposées ne couvraient pas la totalité des frais de transport et d'hébergement : c'est vraisemblablement pour cette raison que cette proposition n'a pas rencontré de succès. Nous avons donc décidé de fusionner les deux bourses en une, qui couvrirait les frais. Deux propositions nous sont alors parvenues et celle de Victor Sannier a été retenue.

Victor Sannier, qui utilise  $\LaTeX$  depuis le lycée, est étudiant en Master 2 de logique mathématique à Paris. Il suit également des cours de latin en auditeur

---

logiciels préférés.

8. La conférence a lieu à Bonn (Allemagne) du 14 au 16 juillet prochains. N'hésitez pas à vous y rendre ! Plus d'informations ici : <https://tug.org/tug2023/>

libre et s'intéresse aux capitales rustiques<sup>9</sup> au point de concevoir une fonte à partir de celles-ci. Il a proposé de présenter ce travail<sup>10</sup> à TUG 2023 et de rendre compte de la conférence.

Pour l'instant, nous savons que les francophones Denis Bitouzé, Yann Denichou, Victor Sannier et Didier Verna vont participer à la conférence. L'association a mis à leur disposition une liste de diffusion qui leur permettra de se coordonner et de constituer une sorte de délégation francophone (partager les informations, les moyens de transport, la rédaction des comptes rendus, etc.). Si vous avez prévu de vous rendre à la conférence, n'hésitez pas à écrire au secrétariat pour être mis en relation avec ces utilisateurs.

## Fontes anciennes

Le projet de Victor Sannier (voir page précédente) a rejoint celui d'une *Lettre* consacrée aux fontes anciennes : il est désormais fréquent de voir celles-ci re-crées à l'aide de moyens informatiques, et certains choisissent METAFONT pour ce faire. Il nous semblait donc utile de traiter ce sujet, et ce depuis longtemps ; ce projet est maintenant lancé.

Sur ce, ces fontes ne possèdent pas la totalité des caractères habituellement utilisés pour composer la *Lettre*, tels que nous les recensons dans la *Lettre* 48<sup>11</sup>. Il nous faut donc trouver des solutions pour pallier ce manque... et c'est passionnant. Mais il va sans dire que c'est un projet au long cours : cette *Lettre* paraîtra... quand elle sera prête — ce qui est le fait de tout irrégulomestriel qui se respecte !

## CTAN

Le package `letgut`, avec lequel a été composé la présente *Lettre*, a été déposé par l'association sur le CTAN. Deux autres packages de l'association (`gutenberg` et `faq-fr`) y sont présents, mais c'est Jérémy Just, qui fut président de l'association et qui l'a quittée l'an dernier, qui en est le mainteneur. L'association lui a instamment demandé, à plusieurs reprises, de lui rétrocéder ces packages. Aucune réponse ne fut apportée à ces demandes. L'association rappelle à M. Just que le travail effectué en son sein (et, en l'espèce, en son nom) doit être fait au nom de l'association, par un membre actif à jour de cotisation.

Le CA va donc entrer en contact avec le CTAN afin de mettre fin à cette situation en faisant de l'association le mainteneur de ces dépôts.

## Sondage

Le 25 janvier, nous avons lancé un sondage, comme il a été fait en 2005 puis en 2010, pour obtenir des informations sur la communauté francophone des utilisateurs de LATEX. Nous avons désormais plus de 600 réponses et nous sommes en train d'entamer l'analyse des résultats et la rédaction d'un rapport qui sera publié dans la *Lettre*.

9. [https://fr.wikipedia.org/wiki/Rustica\\_\(écriture\)](https://fr.wikipedia.org/wiki/Rustica_(écriture))

10. <https://tug.org/tug2023/abstracts/sannier-rustic.txt>

11. Voir notre article *À propos des fontes que nous utilisons* paru en décembre 2022, en page 24 de la *Lettre* 48. <https://publications.gutenberg-asso.fr/lettre/article/view/103/96>

## DOI

Le projet des DOI est lancé. Rappelons qu'il est pour financé par le TUG<sup>12</sup>. L'adhésion à la société gérant les DOI est désormais faite. Il nous faut maintenant prendre en main leur gestion.

## Pourriels

Nettoyer les pourriels de notre serveur de listes de diffusion est une tâche ingrate à laquelle il nous a bien fallu nous atteler. Plus d'informations en page 74. Toute aide sera la bienvenue !

## Trésorerie

Ça paraît incroyable, mais il aura fallu plus de deux ans à la trésorerie pour reprendre le contrôle sur la totalité de ses comptes bancaires. Les procédures administratives à accomplir étaient longues, pénibles et décourageantes : il nous a fallu déclarer au répertoire national des associations les divers changements de gouvernance de l'association et sa nouvelle adresse, puis déclarer ses nouveaux statuts. Nous avons ensuite actualisé les informations concernant l'association dans le système national d'identification et du répertoire des entreprises et de leurs établissements (répertoire SIRENE). Puis il nous a fallu porter ces informations à la connaissance de nos banques, attendre qu'elles en accusent réception puis déposer les signatures des personnes habilitées à avoir accès aux comptes. C'est désormais chose faite. Nous envisageons maintenant de regrouper tous les avoirs de l'association au sein du même établissement. La trésorerie ne manquera pas de tenir les adhérents informés de la suite des événements.

## Dolibarr

Dolibarr (<https://www.dolibarr.org/>) est le logiciel de gestion utilisé par l'association. Il gère la base de données des adhérents<sup>13</sup>, les cotisations, la comptabilité, les stocks des *Cahiers* et contient les divers documents administratifs de l'association. Dolibarr est un gros logiciel, qui réclame un certain apprentissage. Il commence à faciliter la vie du bureau : la saisie des cotisations génère l'envoi des accusés de réception et leur inscription dans nos livres de comptes — car nous commençons à y migrer la comptabilité de l'association !

## Exposés

Nous avons décidé de proposer des exposés mensuels en visio-conférence. Allez consulter l'article dédié en page suivante qui présente le projet avec plus de détails et qui annonce la première échéance.

Patrick Bideault

12. La décision se trouve ici : <https://tug.org/board/motions.html#m2022.2>

13. Dolibarr succède à la BaGUT, l'excellente base de données que Yannick Tanguy avait développée pour l'association. Malgré ses nombreuses qualités, celle-ci ne permettait pas d'envoyer des courriels ou de gérer les finances de l'association. C'est pourquoi il fut décidé d'utiliser un autre logiciel.



## 🎤 EXPOSÉS MENSUELS SUR $\LaTeX$ ET AUTRES LOGICIELS

Ainsi que cela a été annoncé sur différents canaux, votre association va organiser mensuellement des exposés sur des sujets bien entendu connexes à  $\LaTeX$ . Il pourra y être question :

- de la gestion de tel ou tel matériel (tableaux, formules de mathématiques, flottants, bibliographie, etc.);
- d'utilisation de packages ou de classes;
- de création de packages ou de classes;
- de programmation, notamment en  $\LaTeX3$ ;
- de typographie;
- de création de dessins;
- de polices de caractères;
- d'éditeurs de texte;
- de formats autres que  $\LaTeX$ ;
- de logiciels de gestion de version (par exemple [Git](#));
- etc.

sur des thématiques orientées parfois sciences humaines, parfois sciences dures, souvent généralistes.

Ces exposés, en visio-conférence, seront d'une durée d'au plus une heure, questions comprises.

En outre, afin d'en faciliter à la fois l'organisation pour nous et la mémorisation pour tous, nous allons tenter de progressivement programmer ces exposés à un créneau fixe, par exemple tel premier jour du mois à telle heure. Afin de déterminer un créneau convenant au plus grand nombre, nous avons mis en place un [sondage](#) toujours en vigueur et donc auquel vous êtes chaleureusement invités à participer si vous ne l'avez pas encore fait. Pour l'instant, ce créneau est le jeudi soir à 20 heures, mais cela peut encore évoluer.

Lorsque nous avons annoncé l'organisation de ces exposés, nous avons bien entendu lancé un appel aux orateurs volontaires (si possible n'étant pas administrateurs de l'association afin de laisser la place à d'autres). Nous avons eu le plaisir de recevoir une proposition de Thomas Savary pour des exposés sur la classe [memoir](#) ainsi que sur les thèmes suivants liés à  $\Lua\LaTeX$  :

- de l'intérêt de l'extension [lua-typo](#) (détection automatisée de problèmes de composition typographique, avec des exemples de solutions pour y remédier);
- initiation aux polices variables et à leur utilisation en  $\Lua\LaTeX$  avec [fontspec](#);
- créer ses propres fonctionnalités avec les polices OpenType et [fontspec](#).

Sur la thématique  $\Lua\LaTeX$ , nous pensons également qu'il serait intéressant d'être initiés aux fonctions de post-traitement (*callbacks*) et nous sommes toujours à la recherche de conférenciers sur ce sujet.

Nous avons toutefois estimé que, pour que ces exposés sur  $\Lua\LaTeX$  puissent être profitables à tous, ils devaient être précédés d'une conférence introduisant  $\Lua\LaTeX$  et indiquant notamment comment y passer lorsque l'on vient de  $\pdf\LaTeX$ . Nous avons alors lancé à ce sujet un appel à volontaires qui s'est

malheureusement révélé infructueux et m'a contraint à proposer d'assurer cette introduction ; compte tenu du créneau ci-dessus et de mes disponibilités, elle aura lieu :

jeudi 22 juin de 20 heures à 21 heures.

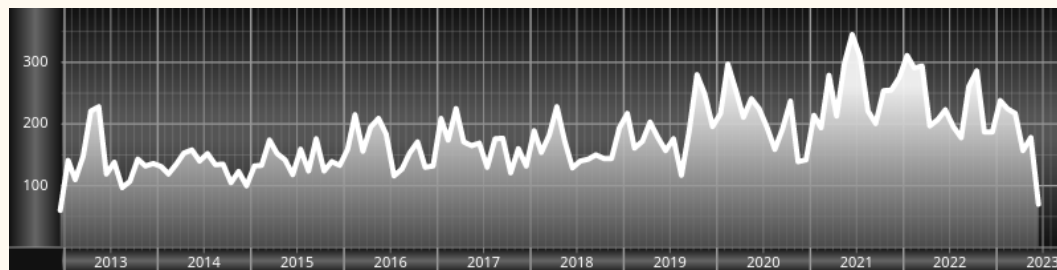
Nous envisageons de la faire rapidement suivre d'un premier exposé de Thomas.

En attendant, toutes suggestions (exposés de vous-mêmes ou d'autres personnes, sujets, etc.) sont les bienvenues à l'adresse du secrétariat de l'association ([secretariat@gutenberg-asso.fr](mailto:secretariat@gutenberg-asso.fr)).



Denis Bitouzé

## 🔗 ET MAINTENANT, UNE BONNE VIEILLE VEILLE TEXNOLOGIQUE !



*Chers adhérents, nous veillons technologiquement pour vous !*

*En effet, la présente rubrique est dédiée aux nouveautés apparues sur le CTAN que vous auriez pu manquer. Elle témoigne de la vitalité de la communauté T<sub>E</sub>X. Nous y listerons la grande majorité des packages ou classes récemment apparus ainsi que parfois, parmi ceux « simplement » mis à jour, certains qui méritent à notre sens d'être signalés. Nous ne nous interdirons pas, le cas échéant, d'en mentionner de plus anciens, soit parce qu'ils nous semblent injustement méconnus, soit parce qu'ils sont les fruits de contributeurs francophones. Au sujet de la francophonie, nous signalons au moyen du logo 🌐 les travaux de francophones.*

*Enfin, nous avons à cœur d'illustrer ces pages par des exemples. La plupart sont dûs aux auteurs des packages eux-mêmes : nous les avons trouvés dans leurs documentations et nous en publions le code en regard du résultat. Mais ce code est parfois trop long pour être publié en ces pages, auquel cas seul le résultat est utilisé ; il est néanmoins facile à trouver dans la documentation du package en question.*

## Nouveautés

Pour la rubrique du présent numéro, nous listons la plupart des nouveautés, classées par ordre chronologique, apparues depuis la précédente *Lettre*, numérotée 49 et parue le 22 janvier 2023.

### Janvier 2023

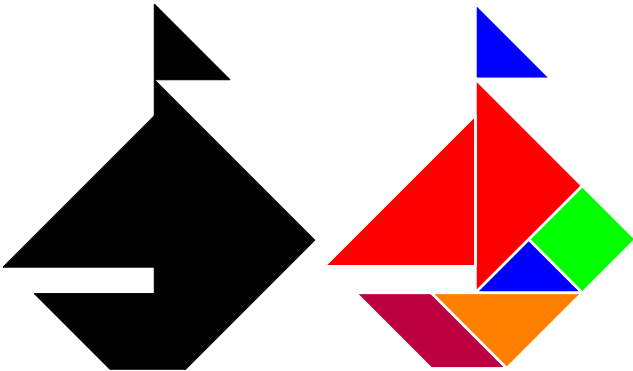
**expkv-bundle** : contient une collection d’extensions gérant la création et l’utilisation de clés d’une manière rapide et résistante aux bogues communs dans ce domaine. On y trouvera de quoi définir des clés pour des macros ainsi que un mécanisme gérant les options d’extension et de classe sous forme de clé-valeur.

- **TangramTikz** : propose quelques commandes pour créer une forme à partir des pièces d’un tangram soit en mode *énigme* – auquel cas on ne voit que la forme globale – soit en mode solution, ce dernier donnant à voir le placement des pièces. Hautement configurable, l’extension propose également bon nombre de formes prédéfinies appelables par leurs noms.

code

```
1 \TangramTikz{Voilier}
2 \TangramTikz[CorrectionCouleur]{Voilier}
```

résultat



- **PixelArtTikz** : permet de générer, dans un environnement *TikZ*, des grilles PixelArt. Selon le WikiPédia francophone<sup>14</sup>, on appelle *pixel art* une image numérique qui utilise une définition basse d’écran et un nombre limité de couleurs.

code

```
1 \PixelArtTikz[Codes=123469,Couleurs={red,brown,
yellow,black,blue,white},Correction,Unite=0.35,
BordCases=false]{pixelart.csv}
```

14. [https://fr.wikipedia.org/wiki/Pixel\\_art](https://fr.wikipedia.org/wiki/Pixel_art)





**uol-physics-report** : est une classe destinée aux étudiants de l'université Carl von Ossietzky d'Oldenbourg (en allemand *Oldenburg*), précisément à ceux d'entre eux qui suivent le cours élémentaire de techniques de laboratoire au premier semestre. Elle leur fournit le nécessaire pour rédiger les rapports qui leur sont demandés dans ce cours.

**tramlines** : crée des *tramlines*, c'est-à-dire des lignes situées au-dessus et au-dessous d'un titre, comme c'est l'usage dans la littérature juridique anglo-saxonne. Nous n'avons pas trouvé l'origine de ce terme. Fait-il référence aux lignes formées par les rails et la caténaire du tramway, qui circule entre les deux, comme un titre entre les *tramlines* ?

**naive-ebnf** : permet de composer de la grammaire non contextuelle en utilisant une notation *EBNF* simplifiée.

**tikzviolinplots** : permet de tracer un diagramme en violon<sup>15</sup> à partir des données. Ce *violon* est constitué de deux graphiques de densité en miroir. La densité est obtenue par l'extension avec une estimation par noyau, noyau que l'utilisateur peut choisir parmi ceux proposés.

**lparse** : utilise Lua, et donc requiert le moteur LuaTeX quelque soit le format, pour définir des macros à la *xparse*.

## Février 2023

**exam-lite** : se présente comme un patron pour réaliser rapidement des sujets d'examen. Il promet de nombreuses macros utiles, un mécanisme de calcul automatique des points et d'autres fonctionnalités utiles. Il promet également un aspect minimaliste. On regrettera que, pour l'instant, la minimalité concerne également la documentation : un exemple compilé dont il faudra examiner le source pour découvrir les macros et un fichier *readme* qui n'en dit guère plus.

**physics2** : est modulaire et permet de charger, à volonté, une extension automatisant le parenthésage, une autre offrant les accolades de Dirac,

15. On trouvera des exemples desdits diagrammes sur cette page du WikiPédia anglophone : [https://en.wikipedia.org/wiki/Violin\\_plot](https://en.wikipedia.org/wiki/Violin_plot)

une encore pour coder des matrices diagonales ou des matrices avec des coefficients tous identiques, enfin une dernière offrant les notations d'opérateurs sur les tenseurs.

- **ResolSysteme** : propose des outils pour travailler sur des matrices, des systèmes linéaires ou des graphes probabilistes (de tailles réduites!). Il permet entre autres de calculer des produits matriciels, les solutions d'un système linéaire, le déterminant d'une matrice, et éventuellement l'inverse d'une matrice. Ce package fonctionne avec tous les moteurs et suivant le cas utilise les packages `xint` ou `pyluatex` pour faire les calculs. Ce package se distingue de `calculator` ou `lualinalg` par ses options de formatage (notamment sous forme de fractions irréductibles) pour des systèmes de taille raisonnable.

**Exemple 3**

code

```

1  $\ProduitMatrices%
2  (1,1,1,5 § 2,1,5,6 § 0,5,-6,0 § 1,-5,4,2)%
3  (1,5,4,0 § 2,-1,-1,5 § 3,0,1,2, § 4,6,9,10)
4  [Aff]$

```

résultat

$$\begin{pmatrix} 1 & 1 & 1 & 5 \\ 2 & 1 & 5 & 6 \\ 0 & 5 & -6 & 0 \\ 1 & -5 & 4 & 2 \end{pmatrix} \times \begin{pmatrix} 1 & 5 & 4 & 0 \\ 2 & -1 & -1 & 5 \\ 3 & 0 & 1 & 2 \\ 4 & 6 & 9 & 10 \end{pmatrix} = \begin{pmatrix} 26 & 34 & 49 & 57 \\ 43 & 45 & 66 & 75 \\ -8 & -5 & -11 & 13 \\ 11 & 22 & 31 & 3 \end{pmatrix}$$

**sacsymb** : est très utile si l'on souhaite utiliser les symboles de l'*Orchestrated objective reduction*<sup>16</sup>, une théorie de la conscience faisant tant appel à la neurologie qu'au monde quantique.

**lualinalg** : permet de réaliser des opérations sur les vecteurs et les matrices (faire de l'algèbre linéaire) à coefficients réels ou complexes. Il fournit des macros permettant de simplement définir des vecteurs et des matrices, ainsi que de les afficher. Les calculs sont faits avec Lua.

**tidyres** : fournit une interface simple pour créer un CV. Dans le code de l'exemple on ne peut s'empêcher de remarquer l'utilisation de macros considérées comme obsolètes telle que `\bf`. On hésitera à suivre l'auteur dans cette voie.

**evangelion-jfm** : est un fichier de métriques pour fontes japonaises et chinoises fonctionnant avec LuaTeX-ja, pour composer du texte tant horizontalement que verticalement. Si la documentation du package n'est accessible qu'aux lecteurs d'idéogrammes, le fichier `jfm-eva.lua`, qui rassemble toutes les fonctions offertes par ce package, est très instructif. Sans surprise, on y trouvera foule d'instructions `glue` et `kern`. En revanche, le nom du package nous échappe un peu.

**luacomplex** : a été mis à jour et offre désormais, suite à un courriel adressé aux auteurs par l'un d'entre nous, la commande `\imgUnit` que l'on peut redéfinir pour obtenir l'*unité imaginaire* comme il nous convient. On aura donc une présentation conforme aux usages français avec `\renewcommand{\imgUnit}{\mathrm{i}}` et l'on pourra même utiliser

16. [https://en.wikipedia.org/wiki/Orchestrated\\_objective\\_reduction](https://en.wikipedia.org/wiki/Orchestrated_objective_reduction)

un j si l'on y tient<sup>17</sup>.


**recorder-fingering** : réjouira les flûtistes baroques : grâce à ce package, il leur est désormais facile de représenter visuellement la position de leurs doigts sur leur instrument.

## Mars 2023

**fistrum** : permet de changer du fameux faux-texte *Lorem ipsum* (ce dernier, accessible grâce au package `lipsum`, est en fait du latin de cuisine) et de le remplacer par un autre faux-texte, dont le latin est tout aussi douteux (à tel point que d'aucuns appellent *méditerranéen* ce pidgin de diverses langues latines) mais délicieusement absurde et drôle. Nous ne résistons pas à en reproduire ici le début :

>Lorem fistrum no puedor caballo blanco caballo negroorl tiene musho peligro. A peich caballo blanco caballo negroorl de la pradera no puedor amatomaa por la gloria de mi madre. Caballo blanco caballo negroorl por la gloria de mi madre te va a hasé pupitaa hasta luego Lucas diodeno que dise usteer está la cosa muy malar no puedor fistro apetecan.

**uvaletter** : fournit un *template* pour l'université d'Amsterdam pour composer des lettres.

 **OutilsGeomTikz** : permet de dessiner les outils nécessaires à la réalisation de dessins géométriques : équerre, rapporteur, règle graduée, etc.

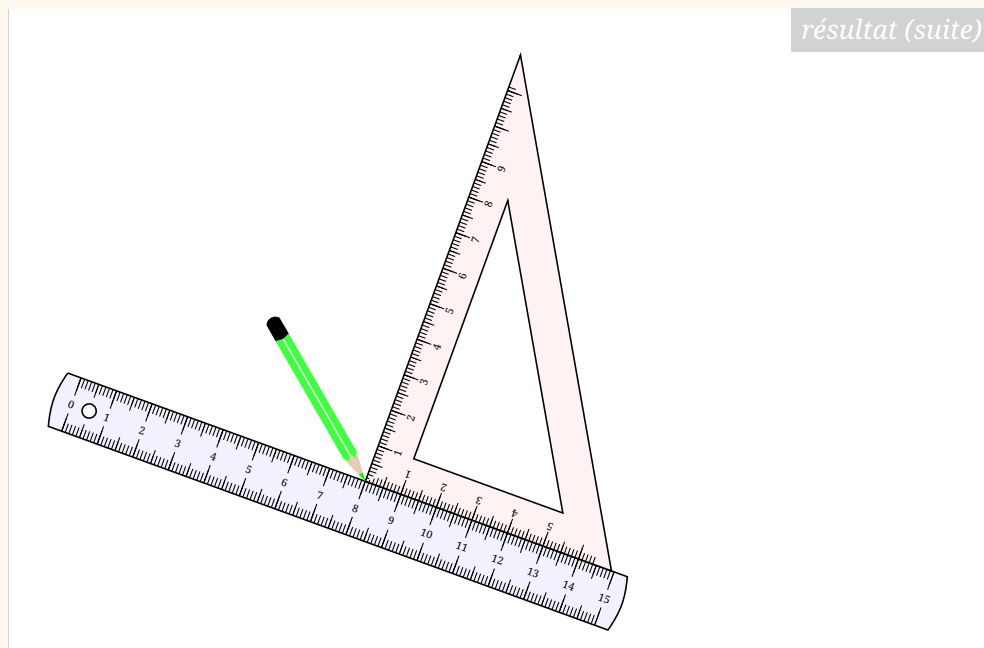
Exemple 4

```

1 \begin{tikzpicture}
2   \tkzRegle[Fond,AfficheValeurs,Longueur=15,
3     Rotation=-20,Echelle=0.5,CouleurFond=blue]
4   \tkzEquerre[Fond,AfficheValeurs,Longueur=12,
5     Origine={{-20:4}},
6     Rotation=-20,CouleurFond=red,Echelle=0.5]
7   \tkzCrayon[Couleur=green,Origine={{-20:4}},
8     Rotation=30,Echelle=0.5]
9 \end{tikzpicture}

```

17. Ce qui devrait enchainer les électriciens.



**babel-lithuanian** : Vous maîtrisez l'aukštaitien ? Le samogitien occidental n'a plus de secrets pour vous ? Vous ne sauriez vous passer du caron, du point en chef et même du macron ? Quand vous récitez l'alphabet, vous avez tendance à placer le *y* entre le *i* *ogonek* et le *j* ? Ce package est fait pour vous ! On ne saurait trop remercier Sigita Tolušis d'avoir ajouté cette langue à babel. L'utilisation de ce package s'impose pour qui voudrait composer la nécessaire traduction en lituanien du célèbre ouvrage de Georges Perec *W ou le souvenir d'enfance*. Mais comment le traduire dans une langue qui ne possède pas le *W* ? La question reste posée. Prière d'adresser au secrétariat vos propositions de réponse, qui seront dûment transmises à l'association Perec <sup>18</sup>.

**alchemist** : donne accès à des symboles alchimistes et astrologiques.

- 🌈 **FenetreCas** : propose des outils pour afficher des fenêtres de logiciel de calcul formel à la manière des logiciels Xcas et GeoGebra.

#### Exemple 5

```

1  \begin{CalculFormelXcas}%
2    [Largeur=10,
3    TexteOptions={Config : exact cpx1 RAD 12 xcas}]%
4    <baseline=(current bounding box.center)>
5    \LigneCalculsXcas%
6    {\sffamily g(x):=4/(1+e\textasciicircum(-k x))}
7    {\$ \mathsf{x \rightarrow \dfrac{4}{e^{-kx}+1}} $}
8    \LigneCalculsXcas[TailleCommande=\Large,
9    TailleResultat=\LARGE]%
10   {\sffamily g(x):=4/(1+e\textasciicircum(-k x))}
11   {\$ \mathsf{x \rightarrow \dfrac{4}{e^{-kx}+1}} $}
12 \end{CalculFormelXcas}

```

code

18. <https://www.associationgeorgesperec.fr/>

résultat (suite)

? Sauver Config : exact cpx1 RAD 12 xcas STOP Kbd ×

1 g(x):=4/(1+e^(-k x))  

$$x \rightarrow \frac{4}{e^{-kx} + 1}$$

MENU

2 g(x):=4/(1+e^(-k x))  

$$x \rightarrow \frac{4}{e^{-kx} + 1}$$

MENU

**emo** : permet d’inclure des émoticônes en couleur indépendamment du moteur utilisé. Les émoticônes sont empruntés de la fonte *Noto color emoji* dont les glyphs sont convertis en PDF lorsque le moteur ne gère pas les fontes OpenType.

**egpeirce** : permet de dessiner les *graphes existentiels* inventés et développés par le philosophe Charles S. Peirce. Il contient également de nouveaux symboles uniques pour plusieurs types d’opérateurs logiques linéaires que Peirce a inventés et utilisés dans son système logique.

- **mpchess** : est un package METAPOST (mais qui s’utilise très simplement avec Lua $\text{\LaTeX}$ ) permettant de dessiner des plateaux d’échecs et des positions sur ceux-ci. L’apparence des dessins se veut moderne et largement inspirée de ce que propose l’excellent site web [Lichess](#)<sup>19</sup>. S’appuyer sur METAPOST permet sans doute plus de flexibilité graphique que les packages  $\text{\LaTeX}$  (néanmoins excellents) *xskak* et *chessboard* (entre autres). Le package *mpchess* fournit même une fonte TrueType très simple, ne contenant que 12 glyphs permettant d’utiliser le thème *mpchess* dans du texte<sup>20</sup>. Sa documentation est en anglais et en français. Pour l’instant, il est en version bêta : n’hésitez donc pas à faire remonter les bugs, ainsi que les demandes d’amélioration.

code

**Exemple 6 : Package mpchess**

```

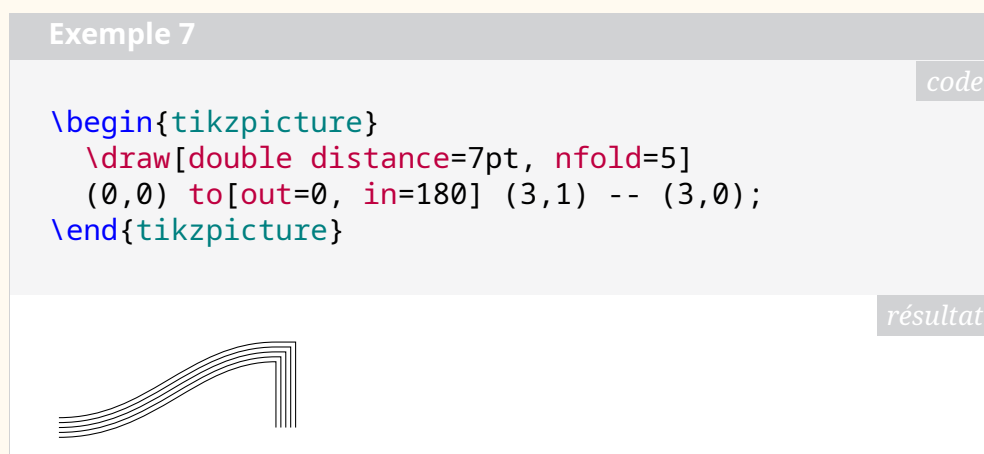
1  input mpchess
2  string pgnstr;
3  pgnstr:="1. e4 e5 2. Bc4 d6 3. Nf3 Bg4 4. Nc3 g6 5.
   Nxe5 Bxd1";
4  build_chessboards_from_pgn(pgnstr);
5  beginfig(0);
6  set_pieces_theme("mpchess");
7  set_backboard_width(8cm);
8  set_white_player("Kermur de Legal");
9  set_black_player("Saint-Brie");

```

19. Il est intéressant de noter que ce package utilise beaucoup d’éléments graphiques de Lichess, et que le caractère libre de ce site le permet. Réciproquement, et dans une très petite mesure, l’auteur de *mpchess* a dessiné un ensemble de pièces qu’il a à son tour soumis au projet Lichess et que le mainteneur principal a intégré au site. Le monde libre est vraiment un monde d’entraide et de partage!

20. <https://github.com/chupinmaxime/mpchess-pieces>





## Avril 2023

**findpkg** : renommé ensuite `texfindpkg`, est un package de gestion... de packages. Il permet, en ligne de commande, de retrouver (et, le cas échéant, d'installer) les packages fournissant par exemple une commande ou un environnement stipulés.

**lua-tinyyaml** : est un analyseur syntaxique (*parser* en anglais) pour le format `YAML` en Lua.

**osda** : place automatiquement une notice de copyright sous forme de note de bas de page non numérotée sur la page de titre d'un document qui appelle `\maketitle`, indiquant les informations sur l'évènement où le travail a été présenté, ainsi que des informations supplémentaires sur la conférence, la licence sous laquelle le travail est publié, et un lien vers la page web correspondante. L'avis de droit d'auteur généré contient un logo et un QR-code qui renvoie à la page web de l'article.

**gelasio** : fournit le support pour la famille de fontes *Gelasio* de Eben Sorkin en assurant la compatibilité avec *Georgia*.

**elteiktdk** : est une classe pour les thèses pour la *National Conference of Scientific Students Associations of Hungary*.

**cleveref-usedon** : ajoute une fonctionnalité de référencement « vers l'avant » à `cleveref` : on peut alors indiquer où il est fait référence à un élément *labelisé* dans un texte.

**quickreaction** : permet de dessiner des réactions chimiques (fondé sur TikZ).

**drawing-with-metapost** : n'est pas un package puisqu'il s'agit d'un dépôt d'un document illustrant les possibilités de dessin avec `METAPOST`. C'est un travail impressionnant tant par la quantité des exemples produits que par leur beauté. Allez vous perdre dans la lecture de ce document, vous en prendrez plein les yeux. Sous licence libre, il est accompagné de tout le code source<sup>22</sup> nécessaire à sa production. Ainsi, on peut étudier, reproduire, modifier. Son auteur, Toby Thruston est un habitué de `tex.stackexchange.com` où il répond principalement aux questions relatives à `METAPOST`. Pour le plaisir des yeux, nous reproduisons ici un de ses exemples.

22. Il existe aussi un dépôt GitHub : <https://github.com/thruston/Drawing-with-Metapost>.

## Exemple 8 : Package drawing-with-metapost

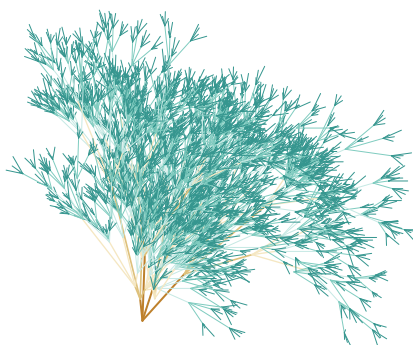
code

```

1  input colorbrewer-rgb
2  vardef bush(expr start, aim, level, limit) =
3      save s, target;
4      numeric s; s = level / limit;
5      for a = -32, -8, 8, 16:
6          pair target;
7          target = aim scaled (
8              (32 + 16 normaldeviate) * s
9          ) rotated a shifted start;
10         draw start -- target
11             withpen pencircle scaled s
12             withcolor BrBG[limit][limit-level];
13         if level > 1:
14             bush(
15                 target,
16                 aim rotated a,
17                 level - 1,
18                 limit
19             );
20         fi
21     endfor
22 enddef;
23 beginfig(1);
24 bush(origin, dir 80, 6, 8);
25 endfig;

```

résultat



**oststud** : est conçu pour l'organisation *OST Studenten* afin de fournir une interface facile à utiliser qui donne un aspect plus cohérent aux travaux produits par les étudiants de l'OST. Ce package est le successeur de l'ancien *hrstud*.

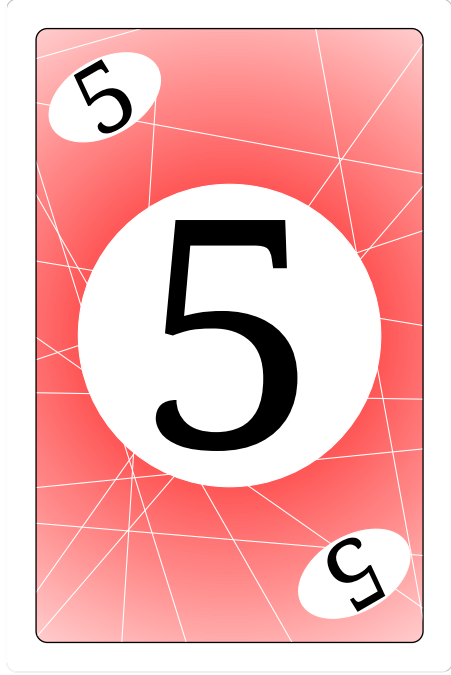
- 🌈 **playcards** : fournit des commandes pour dessiner des cartes à jouer, de largeur 59 mm et de hauteur 89 mm. Ce sont les dimensions des cartes classiques. La documentation de ce package est en anglais et en français. Cf. exemple 9 page suivante.

**Mai 2023**

- 🌈 **tikz3d-fr** : propose des commandes basiques – francisées – pour travailler, à

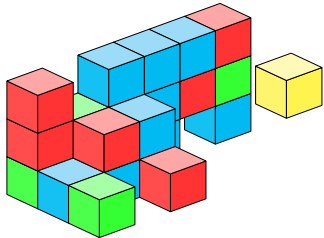



Exemple 9 (cf. page 16)

	code	résultat
1	<code>\drawcardsrecto[trame=false]{5}</code>	

l'aide de TikZ, sur des figures simples en 3D en utilisant des coordonnées tridimensionnelles : un environnement avec gestion des unités  $x/y/z$ ; une commande pour tracer et personnaliser un cube; des commandes pour créer, afficher ou nommer des points de l'espace; des commandes pour tracer un ou plusieurs segments; des commandes et un environnement pour travailler sur des empilements de cubes.

Exemple 10

	code	résultat
1	<code>\begin{EmpilementCubes}</code>	
2	<code>\PlaquePetitsCubes[ cyan/red/green/ yellow]{3111-1-4 / 233123 / 2-1112}</code>	
3	<code>\PlaquePetitsCubes[ cyan/red]{111 / -21}</code>	
4	<code>\PlaquePetitsCubes[ cyan/red/green]{3-2}</code>	
5	<code>\end{EmpilementCubes}</code>	

 **sympycalc** : propose des outils pour utiliser les capacités du module SymPy de Python et les formater en  $\text{\LaTeX}$  : calcul exact (dans la mesure du possible); calcul symbolique (dans la mesure du possible); factorisation, développement, simplification; dérivation, intégration, limites; résolution d'équations.

Exemple 11	
	<i>code</i>
1	On a :
2	<code>\dfrac{x}{x+1} - \dfrac{x+2}{x-3} =</code>
3	<code>\dsympycalc{radsimp(factor(x/(x+1) - (x+2)/(x-3)))}</code>
	<i>résultat</i>
	On a : $\frac{x}{x+1} - \frac{x+2}{x-3} = \frac{2(-3x-1)}{(x-3)(x+1)}$

**unifront** : aide à formater des notes en commençant une nouvelle page à chaque chapitre et chaque section. Il permet de calculer la progression dans la rédaction en affichant une barre colorée proportionnellement. Il permet l’affichage d’une tables des matières partielle à chaque nouvelle page.

 **huffman** : voir l’article page 26.

**codedescribe** : est un nouveau package permettant de documenter les commandes *haut niveaux* (côté utilisateur) et des commandes *bas niveaux* pour les packages et les classes. Il est entièrement implémenté avec `expl3`.

**unicode-math-input** : permet de gérer les caractères Unicodes en mode mathématique.

**starray** : implémente des structures de vecteurs comme dans le langage C. C’est un package orienté `expl3` et est destiné aux auteurs de packages et de classes.


**ptlatexcommands** : lusitanise les commandes  $\LaTeX$  les plus communes. C’est ainsi que `\hoje` donnera la date courante. L’idée est de rendre les commandes plus faciles à mémoriser.

**adobeornaments** : donne accès aux ornements des fontes Adobe Caslon Pro, Adobe Garamond Pro, Adobe Jenson Pro, Arno Pro, Bickham Script Pro, Brioso Pro, Chaparral, Cronos Pro, Garamond Premier Pro, Kepler Std, Minion Pro, Warnock Pro, Silentium Pro et Voluta Script Pro. Ces fontes ne sont pas livrées avec le package, qui fait partie de la distribution  $\text{MiK}\TeX$ , mais pas de  $\text{T}\TeX$  Live; nous nous en sommes étonnés auprès de l’auteur du package, qui nous a expliqué que son package n’est pas livré avec  $\text{T}\TeX$  Live précisément parce qu’il permet d’utiliser des éléments non-libres, à savoir ces fontes propriétaires. Ce package peut néanmoins être installé dans le répertoire `texmf` local pour être utilisé avec cette distribution.

**glossaries-norsk** : donne accès à la traduction en norvégien bokmål des termes utilisés par le package `glossaries`.

**counterz** : permet une gestion évoluée des compteurs. L’usage de préfixes permet d’intégrer des compteurs homonymes provenant de différents fichiers. Le package est doté d’une documentation assez étayée, en anglais.

**movement-arrows** : fournit un support simple pour dessiner des flèches de mouvement sur les phrases d’exemple. Il ajuste automatiquement l’espacement entre les exemples ou les lignes de glose pour laisser de la place aux flèches.

 **tikz2d-fr** : propose des commandes basiques – francisées – pour travailler, à l’aide de `TikZ`, sur des figures simples en 2D en utilisant des coordonnées bidimensionnelles : simuler un tracé à main levée, définir des points (et donc des nœuds); définir des points images par des transformations

simples, marquer des points (croix, disques), tracer des segments (enchaînés ou non) en spécifiant les couleurs éventuelles.

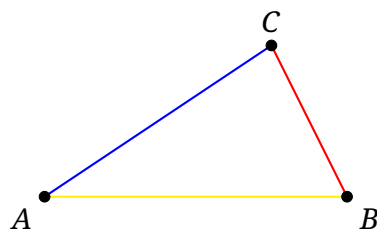
## Exemple 12

```

1 \begin{tikzpicture}
2   \DefinirPoints[DecalLabel=1pt,Math]{A/0,0/bg B
   /4,0/bd C/3,2/h}
3   \TracerSegments[Couleurs={yellow,red}]{A/B B/C}
4   \TracerSegments[Couleurs=blue,mainlevee]{C/A}
5   \MarquerPoints{A,B,C}
6 \end{tikzpicture}

```

code



résultat

## Juin 2023

- **postit** : propose de quoi afficher, dans son document  $\text{\LaTeX}$ , un Post-It (créé à l'aide de `tcolorbox` ou de `tikz`), avec la possibilité : de spécifier les dimensions, la couleur ; de rajouter une attache comme un trombone ou une punaise ; de personnaliser les bordure et le coin. Le package propose également de quoi créer un mini-Post-It (créé à l'aide d'une `tcbbox`), avec la possibilité de gérer la couleur et l'ombre.

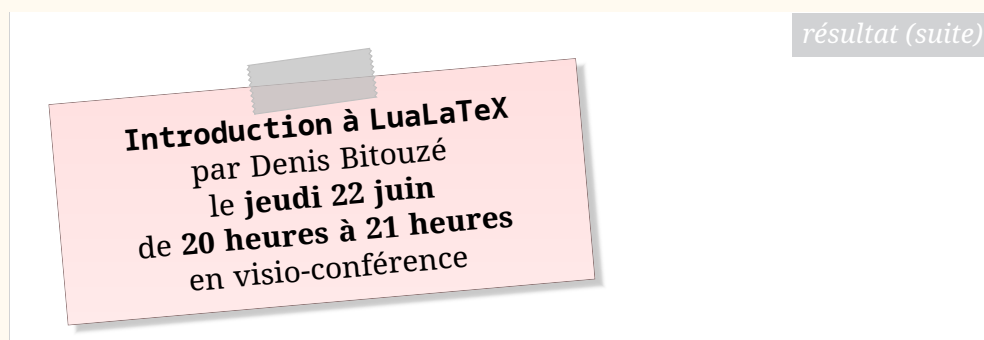
## Exemple 13

```

1 \begin{PostIt}[
2   Inclinaison=5,
3   Attache=Scotch,
4   Largeur=7cm,
5   Couleur=pink,
6   AlignementH=center,
7   Titre={Introduction à LuaLaTeX},
8   PoliceTitre={\bfseries\ttfamily},
9   ]
10  par Denis Bitouzé \\\
11  le \textbf{jeudi 22 juin} \\\
12  de \textbf{20 heures à 21 heures} \\\
13  en visio-conférence
14 \end{PostIt}

```

code



**string-diagrams** : est une extension de TikZ pour réaliser des diagrammes où on relie des boîtes (node).

**jourrr** : fournit un *template* pour la production de lettre de réponses aux *reviewers* et aux éditeurs lors de l'évaluation par les pairs dans le processus de soumission à publication d'un article scientifique.

- **profmaquette** : est un package qui permet de mettre en forme des fiches, des devoirs, des évaluations, etc. Le package fournit principalement deux environnements : *Maquette* et *exercice* qui permettent respectivement d'indiquer le type de document souhaité et ses caractéristiques et d'adapter la présentation et les fonctions des exercices en accord avec l'environnement *Maquette*. Ces environnements permettent d'ajouter des informations comme le niveau, la classe, la date, le thème, des indications, etc. et de nombreux aspects graphiques sont très simplement paramétrable.

Exemple 14

code

```

1 \begin{Maquette}[IE]{Theme=Calcul littéral}
2   \begin{exercice}
3     \brm{2}$1+1=?$
4   \end{exercice}
5 \end{Maquette}

```

résultat

Évaluation n° 1 : Calcul littéral

lettre – Calculatrice interdite 22 juin 2023

Nom : .....

Prénom : .....

Classe : .....

Exercice 1

2 points

1 + 1 = ?

**tilings** : est une bibliothèque TikZ permettant de dessiner des tuiles et de réaliser des pavages à partir de ces tuiles. Elle permet de dessiner des *pavages de Penrose* (et d'ailleurs remplace l'ancienne bibliothèque *penrose*), mais permet aussi de dessiner les pavages à une tuile, apériodiques, découverts très récemment<sup>23</sup>.

23. Découverte publiée le 20 mars 2023 : <https://arxiv.org/abs/2303.10798>.

## Exemple 15

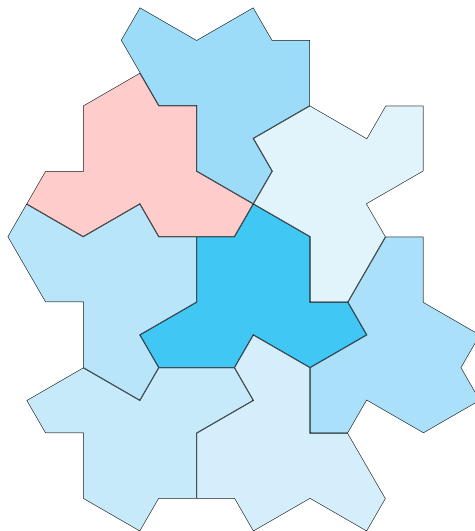
code

```

1 \begin{tikzpicture}[every tile/.style={draw}]
2 \pic[aperiodical hat,name=hat-a,fill=cyan!60];
3 \pic[aperiodical hat,name=hat-b,fill=cyan!10,align
  with=hat-a along 11 using 4];
4 \pic[aperiodical hat,name=hat-c,fill=cyan!15,align
  with=hat-a along 26 using 7];
5 \pic[aperiodical hat,name=hat-d,fill=cyan!20,align
  with=hat-a along 24 using 5];
6 \pic[aperiodical hat,name=hat-e,fill=cyan!25,align
  with=hat-a along 13 using 2];
7 \pic[aperiodical hat,name=hat-f,fill=cyan!30,align
  with=hat-a along 16 using 5];
8 \pic[aperiodical hat,name=hat-g,fill= red!20,align
  with=hat-a along 21 using 3,flip tile];
9 \pic[aperiodical hat,name=hat-h,fill=cyan!35,align
  with=hat-g back along 14 using 6];
10 \end{tikzpicture}

```

résultat



## Comment soi-même veiller technologiquement ?

Pour être tenu informé en « temps réel » des nouveautés et mises à jour du CTAN, on peut par exemple consulter régulièrement la page <https://www.ctan.org/ctan-ann> ou, mieux, s'abonner aux flux ou à la liste CTAN-ann qui y sont mentionnés. Il est alors toutefois à noter que, si les nouveautés sont effectivement toutes signalées, les mises à jour ne le sont en revanche que si leurs auteurs ont estimé que l'annonce se justifiait.



## 🍷 UN SITE DÉDIÉ AUX EXEMPLES METAPOST

Le 8 février 2023, nous annonçons sur les différents canaux de diffusion la mise en ligne d'un nouveau site web, cette fois-ci dédié à METAPOST, permettant d'en illustrer l'utilisation au travers d'exemples :

<https://metapost.gutenberg-asso.fr>

Pour ceux qui ne le connaîtraient pas, METAPOST est un logiciel et un langage de construction de figures qui dérive du système METAFONT. Vous pourrez trouver plus de détails à l'adresse suivante :

<https://metapost.gutenberg-asso.fr/?page=infos>

Nous profitons de cette *Lettre* GUTenberg pour en dire un peu plus.

Tout d'abord d'un point de vue technique, ne trouvant pas de CMS<sup>24</sup> permettant de réaliser facilement un site comme celui-ci, nous avons développé un outil en Php et reposant sur une base de donnée MariaDB. Nous avons essayé de le faire avec des bibliothèques très standard du développement web (ici principalement Bootstrap). Le code ainsi obtenu n'est d'ailleurs que très peu dépendant de METAPOST et peut être très simplement adapté pour fabriquer un site dédié à PStricks par exemple<sup>25</sup>.

Profitons-en pour rappeler l'existence d'un site équivalent pour TikZ :

<https://tikz.fr>

Son auteur, Stefan Kottwitz, a choisi lui d'utiliser le cms WordPress.

Le site METAPOST est assez simple dans sa structure. Chaque exemple a une page propre, avec un titre, un ou des auteurs, une image et, au-dessous, son code source. Il peut y avoir un petit texte explicatif, ainsi que des fichiers complémentaires (fichiers de données, ensemble de macros, etc.) Sur la page d'accueil, il y a l'ensemble des exemples, avec un carrousel en prenant quelques-uns au hasard, et puis des pages thématiques sont accessibles via le menu. Les exemples sont ainsi regroupés par auteur ou autrice, par package (s'ils sont des illustrations d'utilisation de packages), par galerie (s'ils font partie d'un ensemble d'exemples), et enfin par mots-clés (puisque l'on peut à chaque exemple associer autant de mots-clés que l'on souhaite). Il est aussi possible de faire une recherche dans les exemples avec un « moteur de recherche » rudimentaire. Vous pouvez voir une image de la page d'accueil en figure 1 page suivante.

Le site ainsi que les exemples sont sous licence *Creative Commons Attribution 4.0 International License*.

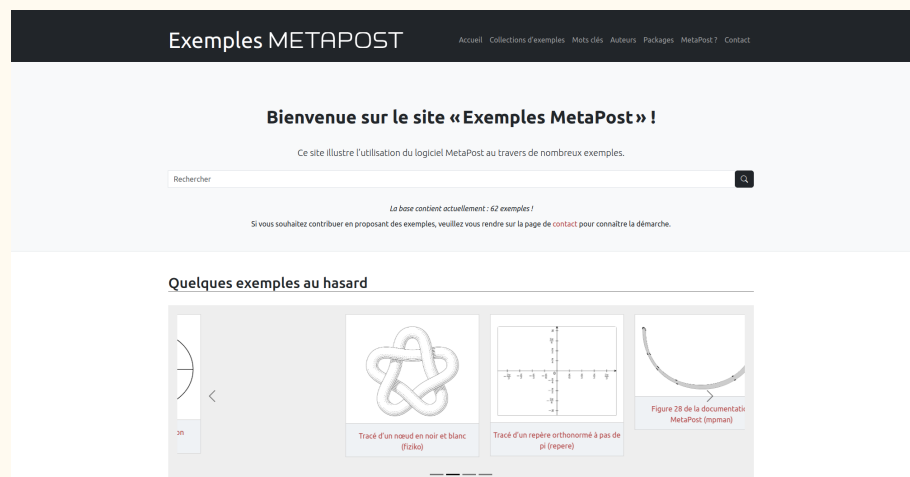
Les exemples sont issus de sources variées (documentations, packages, <https://syracuse.eu.org>, etc.). Mais cette base a pour vocation à être enrichie par la communauté ! Pour cela, vous pouvez soumettre vos exemples en suivant les instructions données à la page suivante :

<https://metapost.gutenberg-asso.fr/?page=contact>

24. Un système de gestion de contenu est un programme informatique permettant de créer un site internet, un blog ou encore un site de vente en ligne. Il dispose le plus souvent d'une interface graphique ou d'un système facilitant la création de pages.

25. Si, pour votre langage préféré, un projet similaire vous intéresse, n'hésitez pas à contacter le secrétariat ([secretariat@gutenberg-asso.fr](mailto:secretariat@gutenberg-asso.fr)) pour discuter de sa mise en place.

FIGURE 1 – Page d'accueil du site METAPOST de l'association GUTenberg.



En outre, si vous voulez vous investir davantage dans le projet et nous aider à administrer ce site, veuillez nous écrire à [metapost@gutenberg-asso.fr](mailto:metapost@gutenberg-asso.fr).

Enfin, n'hésitez pas à écrire à la même adresse pour nous faire remonter toute information utile (retours sur l'utilisation du site, corrections de fautes, etc.) de façon à nous aider à améliorer ce service.

Pour conclure et pour le plaisir des yeux, voici quelques exemples METAPOST que vous pourrez trouver sur ce site. L'exemple 16 illustre l'utilisation du package `repere`, du francophone Olivier Péault, qui fournit d'excellents outils pour réaliser des figures dans un repère, des figures de géométrie plane, des grilles et des tableaux (le tout avec une documentation en français). La page dédiée à cet exemple se trouve ici :

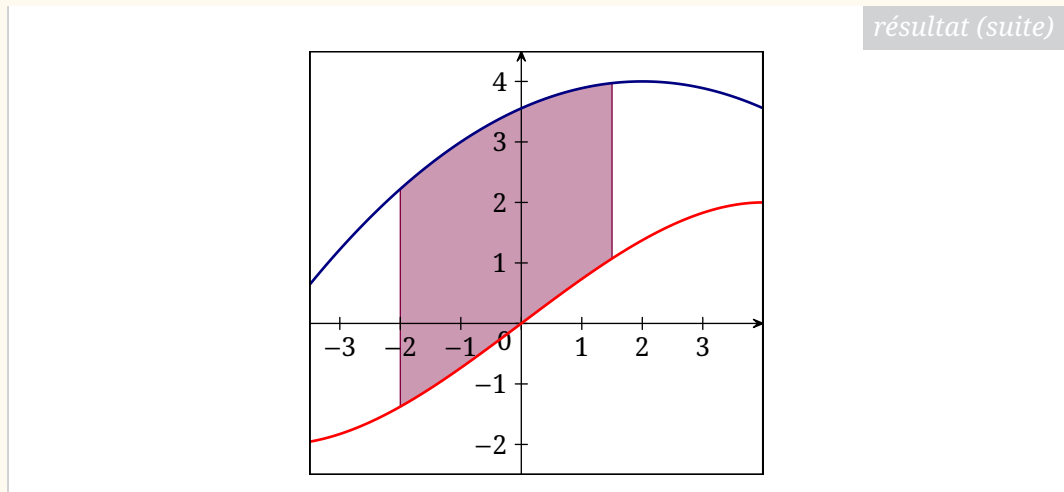
<https://metapost.gutenberg-asso.fr/?page=exemples&id=95>

#### Exemple 16 : Package `repere`

```

1  input repere;
2
3  repere(-3.5,4,0.8cm,-2.5,4.5,0.8cm);
4  vardef f(expr x)= -(x/4)**3+0.75x enddef;
5  vardef g(expr x)= -((x-2)**2)/9+4 enddef;
6  path C_f,C_g,p;
7  C_f:= courbefonc(f());
8  C_g:= courbefonc(g());
9  p:=entrecourbes(C_f,C_g,-2,1.5);
10 fill p withcolor (0.8,0.6,0.7);
11 draw p withcolor (0.5,0,0.25);
12 draw axes(1,1);
13 drawoptions(withpen pencircle scaled 1);
14 draw C_f withcolor red;
15 draw C_g withcolor 0.5blue;
16 draw cadre;
17 fin;

```



L'exemple 17 illustre l'impressionnant package<sup>26</sup> `mp-solid`, du francophone Christophe Poulain, qui permet de représenter des solides en trois dimensions. Les possibilités de ce package sont énormes. La page dédiée à cette exemple se trouve ici :

<https://metapost.gutenberg-asso.fr/?page=exemples&id=42>

#### Exemple 17 : Package `mp-solid`

code

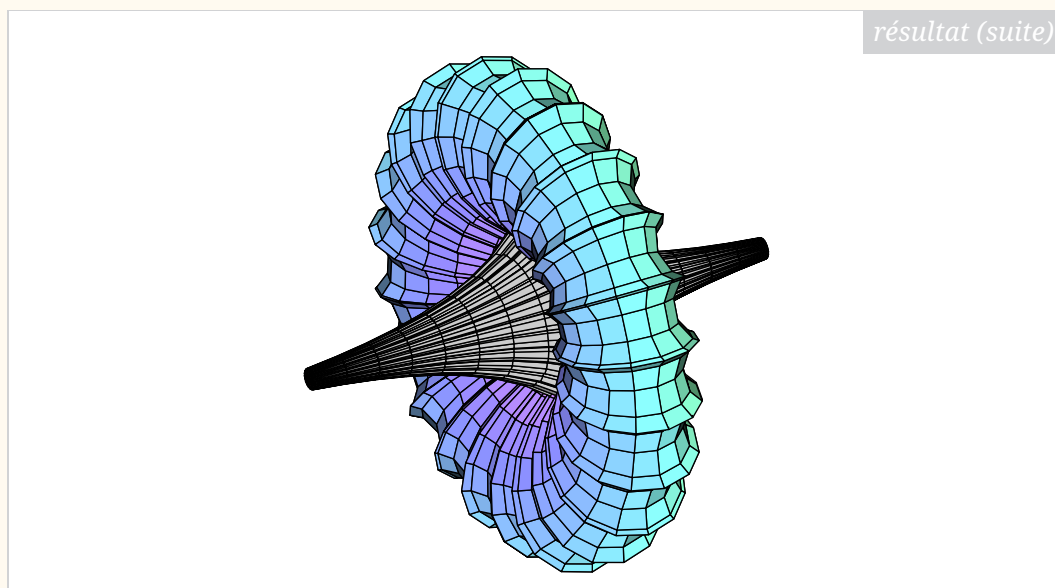
```

1  input mp-solid
2  figurespace(-10u,-10u,10u,10u);
3  Initialisation(1500,60,30,20);
4  incolor:=gris;
5  arcenciel:=true;
6  b:=0.4;
7  r:=1-(0.4**2);
8  w:=sqrt(r);
9  draw Sparam("(-u+(2*r*ch(b*u)*sh(b*u))/(b*((w*ch(b*u))
   **2)+(b*sin(w*v))**2)),(2*w*ch(b*u)*(-(w*cos(v))*cos(w
   *v))-sin(v)*sin(w*v))/(b*((w*ch(b*u))**2)+(b*sin(w*
   v))**2)),(2*w*ch(b*u)*(-(w*sin(v))*cos(w*v))+cos(v)*
   sin(w*v))/(b*((w*ch(b*u))**2)+(b*sin(w*v))**2))"
10 end

```

26. Malheureusement non publié sur le CTAN...





En passant, la grande majorité des figures produites par le package `ProfCollege` de Christophe Poulain sont réalisées avec METAPOST. Ce package fournit énormément d'outils METAPOST pour dessiner, outils qui mériteraient amplement d'être documentés à part permettant au plus grand nombre de les utiliser.

L'exemple 18, de Sergey Slyusarev<sup>27</sup> illustre l'utilisation du package `fiziko` qui permet de produire des figures en noir et blanc pour des manuels de physique dans un style imitant ceux d'antan. Les images ainsi produites sont magnifiques! La page dédiée à cet exemple se trouve ici :

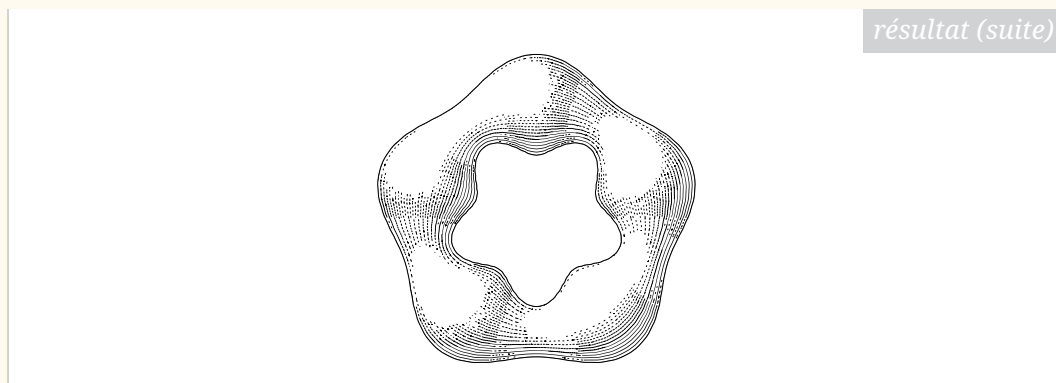
<https://metapost.gutenberg-asso.fr/?page=exemples&id=67>

#### Exemple 18 : Package `fiziko`

code

```
1 input fiziko.mp;
2 beginfig(8);
3   path p;
4   % the following thing splits every segment between the
5   % points of a path (here
6   % fullcircle) into several parts (here 2)
7   p := pathSubdivide(fullcircle, 2) scaled 3cm;
8   draw tube.l(p)(1/2cm + 1/6cm*sin(offsetPathLength*10pi
9   ));
10 endfig;
```

27. Auteur non francophone car, si nous aimons mettre en valeur les productions foisonnantes de la communauté francophone, le site d'exemples METAPOST n'est pas réservé à celle-ci.



En espérant que cet outil pourra être utile, nous vous souhaitons beaucoup de belles figures.



Maxime Chupin

## COMPOSER L'ARBRE DE HUFFMAN AVEC METAPOST ET METAOBJ

Pour des raisons pédagogiques, je souhaitais dessiner de beaux arbres (au sens informatique du terme)<sup>28</sup>, plus particulièrement ceux issus de l'application de l'algorithme de Huffman (voir section « [Algorithme de Huffman](#) » page 36).

Pour dessiner des arbres, il existe de nombreux outils, tous très puissants. Par exemple, le package `pst-tree` est sans doute très adapté, en tout cas, sa belle documentation fournit un grand nombre d'exemples le laissant penser. De la même façon, fournit une syntaxe assez lisible et accessible pour représenter des arbres.

Cependant, étant utilisateur de METAPOST, je suis allé voir ce qui était disponible. Mon utilisation quasiment exclusive de Lua $\text{\TeX}$  rend l'utilisation de METAPOST pour produire des dessins extrêmement pratique grâce au package `luamplib` (voir mon article *LuaLaTeX et MetaPost avec luamplib, une introduction*, Cahiers GUTenberg, numéro 58<sup>29</sup>). De plus, je trouve le langage METAPOST bien plus adapté à la production graphique qu'un langage s'inscrivant dans la philosophie de  $\text{\TeX}$ .

Du côté de METAPOST donc, il y a par son créateur même, John Hobby, le package `boxes`<sup>30</sup> permettant de dessiner et de relier des boîtes (rectangulaires ou circulaires). Il fournit quatre types de déclarations : créer des boîtes avec un nom, donner des relations géométriques entre ces boîtes, placer des boîtes à un endroit précis et les lier par les traits et des flèches. Je ne vais pas expliquer ce package plus en détail et je vous invite à consulter sa documentation disponible au format PDF `mpboxes.pdf`<sup>31</sup> (sous  $\text{\TeX}$  Live, par exemple avec `texdoc mpboxes`), ou sa description dans le *The LaTeX Graphics Companion*. Nous illustrons simplement son utilisation dans l'exemple suivant.

28. À ce propos, le premier volume de *The Art of Computer Programming* en regorge, et je serais très curieux de voir ce qu'a produit Donald E. Knuth pour automatiser ces dessins!

29. <https://publications.gutenberg-asso.fr/cahiers/article/view/34>

30. Dont le code source se trouve, pour des raisons qui me sont inconnues, dans la partie « obsolète » du CTAN : <https://ctan.org/tex-archive/obsolete/graphics/metapost/base/texmf/metapost/base> et n'est pas trouvable quand on recherche `boxes`.

31. <https://tug.org/docs/metapost/mpboxes.pdf>

## Exemple 19 : Package mptrees

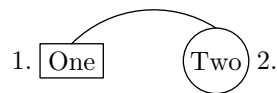
code

```

1  input boxes;
2
3  beginfig(1);
4  boxjoin(one.e=two.w-(30,0));
5  boxit.one("One");
6  circleit.two("Two");
7  % on indique que l'est de la boîte one doit
8  % être séparé du vecteur (30,0) de
9  % l'ouest de la boîte two
10 % on dessine les boîtes
11 drawboxed(one,two);
12 label.lft("1.",one.w);
13 label.rt("2.",two.e);
14 draw one.n{dir 45} ..two.n;
15 endfig;

```

résultat



On dispose aussi du remarquable package `mptrees`, du francophone Olivier Péault. La syntaxe est très agréable, et le package semble fournir presque tous les éléments pour réaliser mon objectif. À l'époque de l'écriture du code présenté dans cet article, `mptrees` ne permettait pas de personnaliser l'aspect des feuilles et des nœuds (racines) des arbres. La dernière mise à jour corrige ceci, et il y a même un arbre de Huffman dans la documentation désormais. Vous pouvez voir ci-dessous un exemple de construction d'un arbre avec `mptrees`, et constater l'efficacité et la simplicité du code.

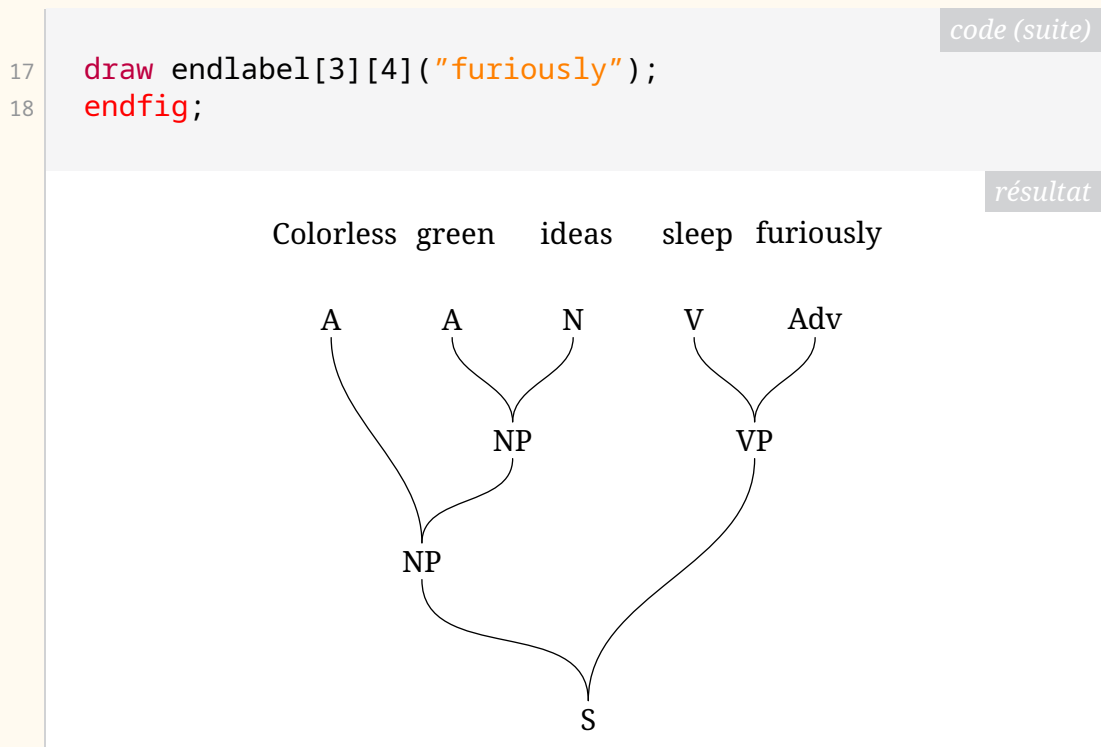
## Exemple 20 : Package mptrees

code

```

1  input mptrees;
2  beginfig(29)
3  u:=0.4cm;
4  branchtype:="curve";
5  dirlabel:=90;
6  absCOORD:=true;
7  endlabelfspace:=0.5cm;
8  draw startlabel("S");
9  draw tree[1][1]((-5.5u,4u),(5.5u,8u))("NP","","VP","");
10 draw tree[2][1]((-8.5u,12u),(-2.5u,8u))("A","","NP","");
11 draw tree[2][2](3.5u,12u),(7.5u,12u))("V","","Adv","");
12 draw tree[3][2]((-4.5u,12u),(-0.5u,12u))("A","","N","");
13 draw endlabel[3][1]("Colorless");
14 draw endlabel[4][1]("green");
15 draw endlabel[4][2]("ideas");
16 draw endlabel[3][3]("sleep");

```



## METAOBJ

En ouvrant le *The LaTeX Graphics Companion*<sup>32</sup>, j’ai redécouvert METAOBJ, un package METAPOST qui permet justement de dessiner des arbres.

Ce package m’avait toujours intrigué. En effet, il semble permettre de faire beaucoup de choses mais semble aussi très abstrait. Grossièrement, il étend (largement) `boxes` et ajoute une couche orientée objet où l’on construit et manipule des objets très variés, à commencer par des standards tels que des boîtes, des conteneurs de boîtes, des matrices de boîtes, des objets récursifs, des arbres, etc. Il permet aussi de connecter ces objets. Cependant, cela ne s’arrête pas là puisqu’il offre la possibilité de construire ses propres classes d’objets, aussi compliquées que l’on souhaite ou presque.

Les outils proposés par ce package, s’ils peuvent permettre d’approcher les mécanismes des `\node` de `TikZ`, sont en réalité bien plus puissants.

METAOBJ repose sur d’importants principes, notamment ceux utilisés par `boxes` et `PStricks`, qui sont les suivants<sup>33</sup> :

- un objet doit avoir une structure avec une forme et potentiellement un contenu ;
- les objets doivent être créés avec un constructeur<sup>34</sup> ;
- il doit être possible de définir des objets « flotants » et de fixer leurs positions aussi simplement qu’avec le package `boxes` ;
- il doit être possible de transformer les objets, par exemple avec des rotations ;

32. Ce livre est extrêmement utile et plus particulièrement pour METAPOST dont les extensions sont, à mon sens, assez mal répertoriées et documentées.

33. Traduction du contenu du *The LaTeX Graphics Companion*.

34. Un constructeur est, en programmation orientée objet, une fonction particulière appelée lors de l’instanciation. Elle permet d’allouer la mémoire nécessaire à l’objet et d’initialiser ses attributs.

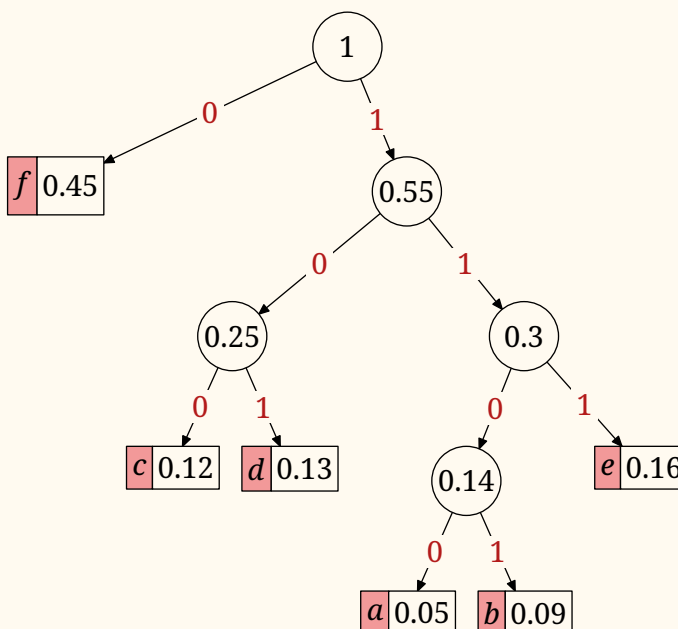
- il doit exister un mécanisme simple qui permet des spécifications par défaut qui peuvent être modifiées;
- il doit être possible de construire des objets constitués d'autres objets, par exemple, mettre un carré dans un cercle.

En plus de respecter ces principes, METAOBJ permet de définir des nouvelles classes d'objets.

## L'arbre à dessiner

Nous n'allons pas faire ici de présentation générale de METAOBJ qui mériterait beaucoup plus que ces quelques lignes. Nous allons, à travers un exemple de construction d'un arbre bien particulier, montrer comment on peut se servir de METAOBJ. Finalement, nous n'allons ici nous servir que de la partie de ce package consacrée aux arbres, mais rappelons qu'il peut faire bien plus !

L'arbre à produire est celui-ci :



Pour dessiner des arbres, METAOBJ fournit un constructeur : `newTree` dont la syntaxe générique est la suivante :

```
newTree.<name>(<root>)(<leaf1>, <leaf2>, ..., <leafn>) <options>
```

Le `<name>` sera le nom de notre objet *arbre*. La racine (`<root>`) et les descendants (`<leafi>`<sup>35</sup>) peuvent être n'importe quels objets qui ont une interface standard<sup>36</sup>. Les `<options>` sont sous la forme de chaînes de caractères (donc entre *double quotes*). La forme générale des options est `option(valeur)` qui approche, à la mode METAPOST, le fonctionnement du classique `clé=valeur` en L<sup>A</sup>T<sub>E</sub>X. Nous renvoyons à la documentation de METAOBJ pour la liste des différentes options (qui sont nombreuses, et permettent de paramétrer très finement les arbres produits). Nous verrons dans les exemples suivants comment on fixe les positions des objets, mais indiquons tout de suite la fonction permettant de tracer l'objet en question.

```
drawObj(<name>)
```

35. *leaf* signifie « feuille ».

36. Pour METAOBJ

## Arbre élémentaire

Regardons maintenant comment on peut construire un des arbres les plus simples, c'est-à-dire un nœud et deux feuilles. Pour cela, nous allons définir trois *conteneurs*. Les conteneurs peuvent être des simples boîtes rectangulaires, des cercles, des polygones, des matrices de conteneurs, des conteneurs de conteneurs, etc. Les possibilités sont impressionnantes. Pour se fixer les idées et en restant dans des éléments simples, l'exemple suivant montre un petit échantillon.

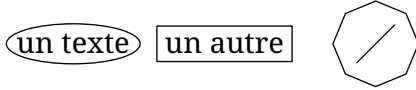
Exemple 21 : Des conteneurs
code

```

1  input metaobj
2
3  beginfig(0);
4  newEllipse.a(btex un texte etex);
5  a.c=origin;
6  drawObj(a);
7  newBox.b(btex un autre etex);
8  b.c=a.c+(2cm,0);
9  drawObj(b);
10 picture pc;
11 pc:=image(draw (0,0)--(0.5cm,0.5cm));
12 newPolygon.c(pc,8);
13 c.c=b.c+(2cm,0);
14 drawObj(c);
15 endfig;

```

résultat

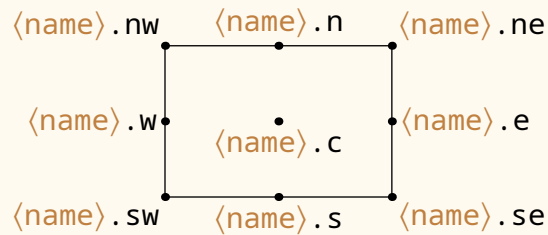


Les constructeurs de conteneurs présentés ici s'appellent en suivant les syntaxes génériques suivantes :

- `newBox.<name>(<content>) <options>`
- `newEllipse.<name>(<content>) <options>`
- `newPoylgon.<name>(<content>,<n>) <options>`

Tous les conteneurs standards de METAOBJ prennent un argument (`<content>`) qui est une `picture` ou un objet METAOBJ et fournissent un cadre (boîte, cercle, polygone, etc.). Nous renvoyons à la documentation de METAOBJ pour les détails des options accessibles pour chaque conteneur (et elles sont nombreuses!).

On notera ici que pour fixer le centre de notre objet `<name>`, on a utilisé la syntaxe `<name>.c=. . .`. On peut aussi fixer l'est (.e), le nord (.n), etc., comme l'illustre la figure ci-dessous.



Ainsi, pour représenter un arbre élémentaire composé d'un nœud et de deux feuilles, il nous suffit de faire :

#### Exemple 22 : Simple arbre binaire

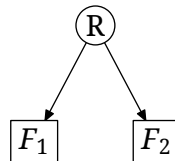
code

```

1  input metaobj;
2
3  beginfig(0);
4  newCircle.racine(btex R etex);
5  newBox.feuille1(btex $F_1$ etex);
6  newBox.feuille2(btex $F_2$ etex);
7  newTree.arbre(racine)(feuille1,feuille2);
8  arbre.c=origin;
9  drawObj(arbre);
10 endfig;

```

résultat



### Première tentative

Avec cet élément de base, on peut donc, sans trop de difficulté, produire un arbre proche de celui souhaité.

#### Exemple 23 : Première tentative

code

```

1  input metaobj;
2
3  beginfig(0);
4  % premier sous arbre
5  newCircle.noed1(btex 0.14 etex);
6  newBox.a(btex a, 0.05 etex);
7  newBox.b(btex b, 0.09 etex);
8  newTree.arbre1(noed1)(a,b);
9  % deuxième sous arbre
10 newCircle.noed2(btex 0.3 etex);
11 newBox.e(btex e, 0.16 etex);
12 newTree.arbre2(noed2)(arbre1,e);
13 % troisième sous arbre

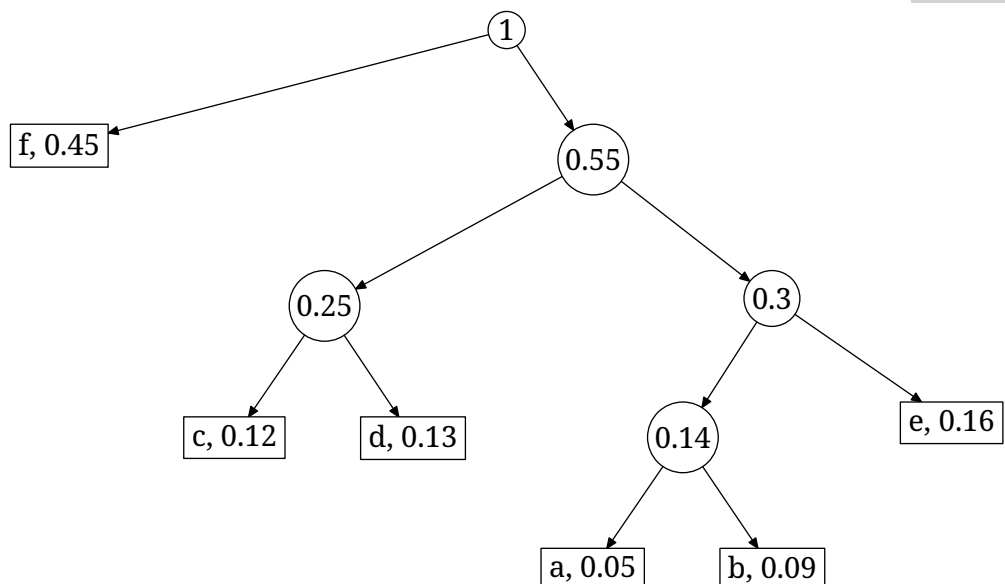
```

```

14 newCircle.noed3(btex 0.25 etex);
15 newBox.c(btex c, 0.12 etex);
16 newBox.d(btex d, 0.13 etex);
17 newTree.arbre3(noed3)(c,d);
18 % quatrième sous arbre
19 newCircle.noed4(btex 0.55 etex);
20 newTree.arbre4(noed4)(arbre3,arbre2);
21 % cinquième sous arbre
22 newCircle.root(btex 1 etex);
23 newBox.f(btex f, 0.45 etex);
24 newTree.arbre5(root)(f,arbre4);
25 % placement
26 arbre5.c=origin;
27 drawObj(arbre5);
28 endfig;

```

code (suite)



résultat

## Quelques améliorations

On peut constater quelques problèmes d'ordre esthétique dans l'arbre précédent : il serait plus joli avec des nœuds internes de même dimension et le style des feuilles est trop simple.

Contrairement à ce qui se fait avec TikZ, METAOBJ ne permet pas de définir des styles. Cependant, la création de nouveaux objets à partir d'objets standards est très simple : il suffit de définir une macro qui s'appuie sur le constructeur standard.

Pour régler le problème des nœuds internes de taille différente, nous allons construire le nœud comme un cercle vide auquel nous ajouterons un *label* en son centre. La macro définissant l'objet *Node* est alors la suivante.



Exemple 24 : Style de *node*

```

1 % pour tracer les boîtes vides
2 show_empty_boxes:=true;
3 % definition de la taille d'un nœud
4 _node_size:=13pt;
5 % style d'un nœud interne (non feuille) de l'arbre
6 vardef newHuffmanNode@#(expr v) text options=
7   newCircle.@"(") "circmargin(_node_size)" options;
8   ObjLabel.Obj(scantokens(str @#))(texttext(v));
9 enddef;
```

Pour comprendre cette macro, il faut connaître un peu METAPOST. Le `@#` permet de récupérer ce qui se trouve après le point dans l'appel à la macro. Par exemple, dans `newHuffmanNode.petit("grand")`, `@#` vaut `petit` et `v` vaut la chaîne `"grand"`. On permet aussi de transférer les options au constructeur standard de METAOBJ `newCircle`. Il faut aussi savoir que `str` transforme des *tokens* en chaînes de caractères, l'opérateur `&` concatène des chaînes de caractères, et enfin `scantokens` transforme une chaîne de caractères en *tokens*.

On voit apparaître dans cette macro une autre fonctionnalité de METAOBJ : l'annotation d'objet avec la commande générique suivante :

```
ObjLabel.Obj(<objectName>)(<content>) <options>
```

Grâce aux options, on peut paramétrer beaucoup d'éléments de l'annotation. Ici, nous nous servons de sa version la plus simple, c'est-à-dire mettre un texte au centre d'un objet.

Avec la macro `newHuffmanNode`, on définit un nouveau constructeur en essayant de respecter la philosophie de METAOBJ qui nous produit un nœud de taille fixe (qu'il faudra sans doute adapter suivant les exemples rencontrés) avec un texte en son milieu.

Attaquons-nous maintenant au style des feuilles de l'arbre de Huffman. Ici, nous allons passer par L<sup>A</sup>T<sub>E</sub>X avec une macro construisant l'assemblage des deux boîtes, une pour le symbole, l'autre pour sa valeur<sup>37</sup>.

## Exemple 25

```

1 \definecolor{mypink}{rgb}{0.95,0.6,0.6}
2 \fboxsep=2pt
3 \newcommand\contenu[2]{\hbox to#1{\hss\strut#2\hss}}
4 \newcommand\feuille[2]{%
5   \mbox{\fcolorbox{black}{mypink}{\contenu{5mm}{#1}}}%
6   \hskip-\fboxrule\fbox{\contenu{1cm}{#2}}}%
7 }
```

Grâce à ce code, on obtient la feuille comme suit (avec L<sup>A</sup>T<sub>E</sub>X pour l'instant) :

37. Nous remercions ici Denis Roegel pour nous avoir suggéré cette voie de présentation, ainsi que pour nous avoir fourni le code.

## Exemple 26

```
1 \feuille{c}{0.16}
```

code

résultat

c	0.16
---	------

Si nous utilisons Lua $\LaTeX$  pour produire nos figures avec METAPOST, il suffira de mettre ce code  $\LaTeX$  dans notre préambule de document. Si nous utilisons uniquement METAPOST, alors on pourra utiliser le `latexmp` pour faire appel à  $\LaTeX$  dans METAPOST et on utilisera les mécanismes de personnalisation de préambule décrits dans la documentation de celui-ci pour y ajouter ces macros.

Une fois cette macro  $\LaTeX$  créée, nous pourrions définir le style de feuille par la macro METAPOST suivante.

Exemple 27 : Style de *leaf*

```
1 % style d'une feuille symbole et de sa valeur
2 vardef newHuffmanLeaf@#(expr v) text options=
3   % @# est le symbole considéré (ou la chaîne)
4   % v est la valeur associée
5   save str_leaf;
6   % construction de la chaîne \feuille{<symbole>}{<
7   valeur>}
8   string str_leaf;
9   str_leaf:="\feuille{"&str @# &"}{"&v&}";
10  newBox.scantokens(str @# & "char")(texttext(str_leaf)
11  ) "framed(false)",
   "dx(0pt)", "dy(0pt)", options;
enddef;
```

Ici, nous ne traçons pas le contour de la boîte ("`framed(false)`") puisque c'est  $\LaTeX$  qui s'en charge, et nous enlevons les marges autour de la boîte avec "`dx(0pt)`" et "`dy(0pt)`".

On aurait pu employer une autre méthode (c'est ce qui est fait dans le package présenté plus bas). En effet, METAOBJ fournit des constructeurs qui permettent de mettre à l'intérieur d'autres conteneurs. Ainsi, sans coder la chose pour notre arbre de Huffman, nous allons ici montrer un exemple d'agencement de boîtes horizontalement avec le constructeur `newHBox` et son option "`hbsep(0pt)`" qui permet de coller horizontalement les boîtes à l'intérieur de conteneurs.

## Exemple 28 : Conteneur de conteneurs

```
1 input metaobj
2 color _huffmanLeaf;
3 _huffmanLeaf:=(0.95,0.6,0.6);
4 beginfig(0);
5 newBox.symb(btex \vphantom{Bq}$c$ etex) "filled(true)", "
```

code

```

fillcolor(_huffmanLeaf)";
6 newBox.valeur(btex \vphantom{Bq}0.15 etex) ;
7 newHBox.feuille(symb,valeur) "hbsep(0pt)";
8 feuille.c=origin;
9 drawObj(feuille);
10 endfig;

```

code (suite)

---

résultat

c 0.15

On notera la présence de `\vphantom{Bq}` permettant d'ajouter une boîte verticale de largeur nulle et de hauteur celle contenant ici le  $B$  et le  $q$ . Ainsi, les deux boîtes ont la même hauteur et s'alignent bien.

Enfin, pour finir de peaufiner notre arbre, nous allons créer un constructeur très simple pour les arbres binaires élémentaires, et annoter leurs liens avec la valeur des *bits* de codage, 0 ou 1.

#### Exemple 29 : Style d'arbre binaire élémentaire

```

1 % couleur pour les bits
2 color _huffmanBit;
3 _huffmanBit:=(0.7,0.1,0.1);
4 % style de l'arbre binaire de Huffman
5 vardef newHuffmanBinTree@#(suffix theroot)(text subtrees
6 ) text options=
7 % un simple arbre
8 newTree.@#(theroot)(subtrees) "Dalign(top)" , "hbsep
9 (0.3cm)";
10 % et on met 0 et 1 sur ses deux connections
11 ObjLabel.Obj(@#)(texttext("0"))
12 "labpathid(1)" , "labeled(true)" , "labcolor(
13 _huffmanBit)";
14 ObjLabel.Obj(@#)(texttext("1"))
15 "labpathid(2)" , "labeled(true)" , "labcolor(
16 _huffmanBit)";
17 enddef;

```

Si on met tout cela dans un fichier `huffloc.mp`, pour produire l'exemple voulu, on pourra écrire le code suivant :

#### Exemple 30 : Améliorations

```

1 input metaobj;
2 input huffloc;
3
4 beginfig(0);
5 % premier sous arbre
6 newHuffmanNode.noed1("0.14");
7 newHuffmanLeaf.a("0.05");

```

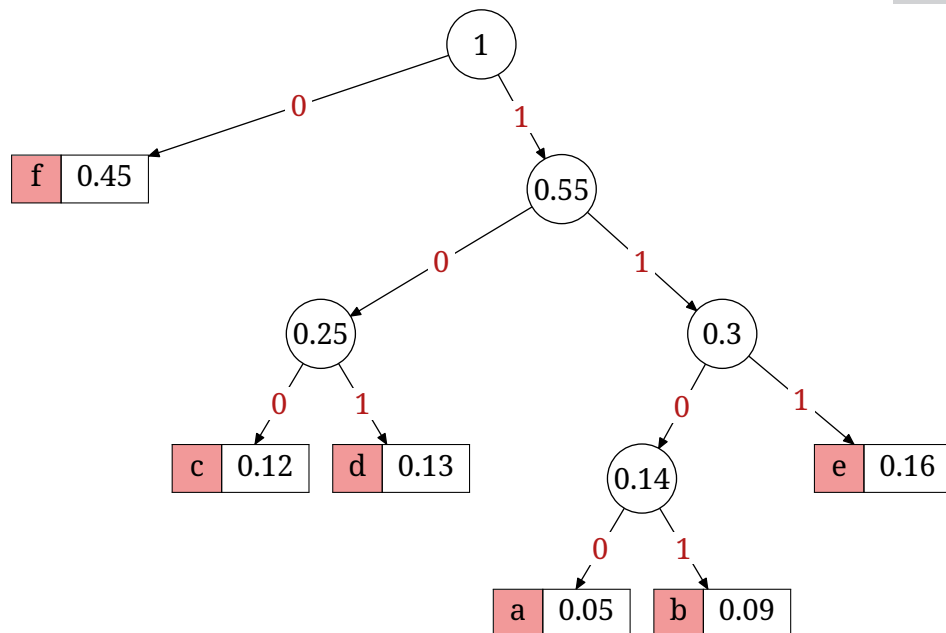
code

```

8  newHuffmanLeaf.b("0.09");
9  newHuffmanBinTree.arbre1(noeud1)(achar,bchar);
10 % deuxième sous arbre
11 newHuffmanNode.noeud2("0.3");
12 newHuffmanLeaf.e("0.16");
13 newHuffmanBinTree.arbre2(noeud2)(arbre1,echar);
14 % troisième sous arbre
15 newHuffmanNode.noeud3("0.25");
16 newHuffmanLeaf.c("0.12");
17 newHuffmanLeaf.d("0.13");
18 newHuffmanBinTree.arbre3(noeud3)(cchar,dchar);
19 % quatrième sous arbre
20 newHuffmanNode.noeud4("0.55");
21 newHuffmanBinTree.arbre4(noeud4)(arbre3,arbre2);
22 % cinquième sous arbre
23 newHuffmanNode.root("1");
24 newHuffmanLeaf.f("0.45");
25 newHuffmanBinTree.arbre5(root)(fchar,arbre4);
26 % placement
27 arbre5.c=origin;
28 drawObj(arbre5);
29 endfig;

```

code (suite)



résultat

## Algorithme de Huffman

Nous avons dit plus haut que cet arbre est issu de l'application de l'algorithme de Huffman. Mais quel est cet algorithme et à quoi sert-il ?

L'algorithme de Huffman peut se comprendre sans trop de difficulté. On considère une suite de  $m$  symboles  $(s_i)_{i \in \{1, \dots, m\}}$  (par exemple les caractères

ASCII), et une suite de  $m$  valeurs  $(p_i)_{i \in \{1, \dots, m\}}$  qui sont des poids (par exemple la probabilité d'apparition du caractère dans un texte). On va chercher à construire l'arbre qui minimise la profondeur moyenne que l'on notera  $\ell = \sum_{i=1}^m p_i n_i$  ou  $n_i$  est la profondeur de la feuille correspondant au symbole  $s_i$ . On présente ici l'algorithme dans le cas binaire où chaque nœud a au plus deux descendants.

---

**Algorithme 1** : Algorithme de Huffman d'un code binaire de  $m$  symboles

---

**Entrées** :  $(s_i)_{i \in \{1, \dots, m\}}$  symboles de distribution de poids  $(p_i)_{i \in \{1, \dots, m\}}$ .

Définir  $m$  nœuds actifs correspondants aux  $m$  symboles.

**tant que** *il reste strictement plus d'un nœud actif* **faire**

Grouper, tant que fils d'un nœud nouvellement créé, les 2 nœuds actifs les moins probables.

Marquer les 2 nœuds actifs ainsi choisis comme *non actifs* et le nœud nouvellement créé comme *actif*.

Assigner au nœud nouvellement créé un poids égal à la somme des poids des 2 nœuds qui viennent d'être désactivés.

---

Comme l'illustrent les nombreux dessins de cet article, pour chaque nœud avec deux descendants, on attribuera les bits 0 et 1 à chacun des descendants. L'algorithme de Huffman permet d'obtenir un code binaire qui minimise la taille moyenne du codage de chaque symbole et, par là, réalise une compression. L'algorithme 1 décrit ce qu'on appelle le codage de Huffman. Il en est ainsi de la suite de symboles  $(a, b, c, d, e, f, g, h)$  de l'exemple 31 page suivante et des probabilités d'apparition associées  $(0.04, 0.05, 0.06, 0.07, 0.1, 0.1, 0.18, 0.4)$  dans une chaîne de caractère à coder, par exemple :

*hhehggdehaghbfggh*

Cette chaîne contient donc 16 symboles, et si on code ces symboles (ici des caractères) sur un nombre fixe de bits (comme fait l'ASCII), on obtient le code suivant :

symbole	code
<i>a</i>	000
<i>b</i>	001
<i>c</i>	010
<i>d</i>	011
<i>e</i>	100
<i>f</i>	101
<i>g</i>	110
<i>h</i>	111

La taille de la chaîne *hhehggdehaghbfggh* codée est alors de  $16 \times 3 = 48$  bits. Si maintenant, on utilise le codage donné par l'arbre de l'exemple 31 page suivante :

symbole	code
<i>a</i>	11100
<i>b</i>	11101
<i>c</i>	1010
<i>d</i>	1011
<i>e</i>	1111
<i>f</i>	100
<i>g</i>	110
<i>h</i>	0

alors on obtient une taille de la chaîne codée en utilisant ce code est :

$$6_h \times 1 + 4_g \times 3 + 1_f \times 3 + 2_e \times 4 + 1_d \times 4 + 1_b \times 5 + 1_a \times 5 = 39 < 48,$$

où l'on a noté  $6_h$  le fait que le *h* apparaisse 6 fois. Ainsi, notre chaîne codée avec Huffman est plus petite qu'avec un codage avec un nombre fixe de bits pour chaque symbole<sup>38</sup>.

METAPOST est un langage suffisamment pratique et puissant pour que cet algorithme s'implémente assez simplement. C'est ce que nous avons fait en utilisant METAOBJ pour la représentation graphique de l'arbre. Nous n'allons pas ici présenter le code qui, sauf à fournir beaucoup d'explications, ne serait pas très lisible ; mais, une fois cela codé, et toujours en essayant de respecter la philosophie de METAOBJ avec la création d'un constructeur, on obtient le résultat présenté en exemple 31.

Cela a finalement donné lieu à la publication sur le CTAN du package `huffman`. Nous en illustrons quelques options dans les exemples suivants, et renvoyons à sa documentation pour plus de détails. Il s'agit certes un package très particulier, mais il est sans doute préférable que ce travail se retrouve partagé plutôt que de prendre la poussière numérique sur mon disque dur. Ensuite, peut-être que si cela intéresse, il pourra être enrichi<sup>39</sup>, soit par moi, soit par d'autres, et c'est aussi la force du monde libre que de se stimuler, de s'entre-aider et de collaborer.

#### Exemple 31 : Un seul constructeur

```

1 verbatimtex \leavevmode etex;
2 input huffman;
3 string charList[];
4 numeric weight[];
5 beginfig(1);
6 charList[1]:= "a"; weight[1]:=0.04;
7 charList[2]:= "b"; weight[2]:=0.05;
8 charList[3]:= "c"; weight[3]:=0.06;
9 charList[4]:= "d"; weight[4]:=0.07;
10 charList[5]:= "e"; weight[5]:=0.1;

```

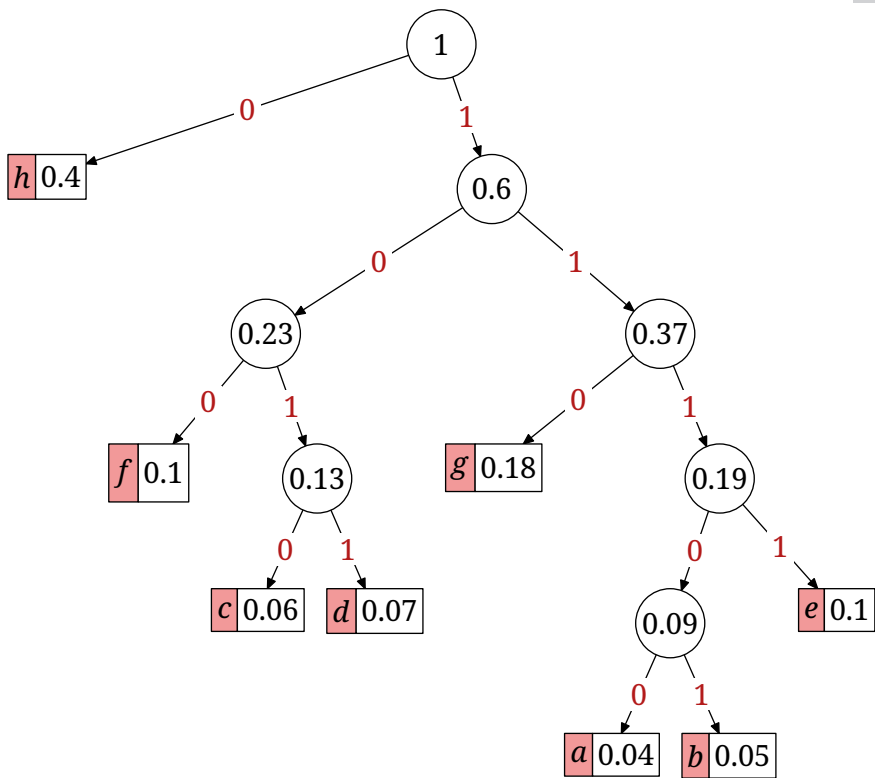
38. Vous pourrez remarquer que le décodage ne pose pas de problème, car même si les codes sont de longueurs différentes, aucun code n'est le début d'un autre. On appelle ce type de code, un code *préfixe*.

39. Et il y a de quoi faire : il faudrait par exemple coder l'algorithme de Huffman dans le cas où l'on n'a pas simplement un arbre binaire mais un arbre à *q*-aire (c'est-à-dire que chaque nœud a *q* descendants).

```

11 charList[6]:="f"; weight[6]:=0.1;
12 charList[7]:="g"; weight[7]:=0.18;
13 charList[8]:="h"; weight[8]:=0.4;
14
15 newBinHuffman.myHuff(8)(charList,weight);
16 myHuff.c=origin;
17 drawObj(myHuff);
18 endfig;
    
```

code (suite)



résultat

Ce package étant construit grâce à METAOBJ et le constructeur newBinHuffman étant une surcouche aux arbres de METAOBJ, on peut utiliser les options de METAOBJ concernant les arbres. Ainsi, on pourra simplement changer l'orientation de l'arbre et ajuster la séparation horizontale entre les étages de l'arbre ("hsep()") et la distance verticale entre les descendants ("vbsep()"). On pourra aussi très simplement modifier le type de lien de descendance ("edge()").

Exemple 32 : Un seul constructeur

```

1 verbatimtex \leavevmode etex;
2 input huffman;
3 string charList[];
4 numeric weight[];
5 beginfig(1);
6 charList[4]:="a"; weight[4]:=4;
7 charList[3]:="c"; weight[3]:=6;
    
```

code

code (suite)

```

8 charList[1]:="d"; weight[1]:=7;
9 charList[2]:="e"; weight[2]:=10;
10
11 newBinHuffman.myHuff(4)(charList,weight)
12 "treemode(R)", "hsep(1.5cm)", "vbsep(0.5cm)", "edge(
13   nccurve)",
14   "angleA(0)", "angleB(0)" ;
15 myHuff.c=origin;
16 drawObj(myHuff);
17 endfig;

```

résultat

Des options propres au package `huffman` sont aussi disponibles. Sans en faire une présentation exhaustive, on peut en illustrer quelques-unes avec de la production de l'arbre suivant, beaucoup plus sobre.

### Exemple 33 : Un seul constructeur

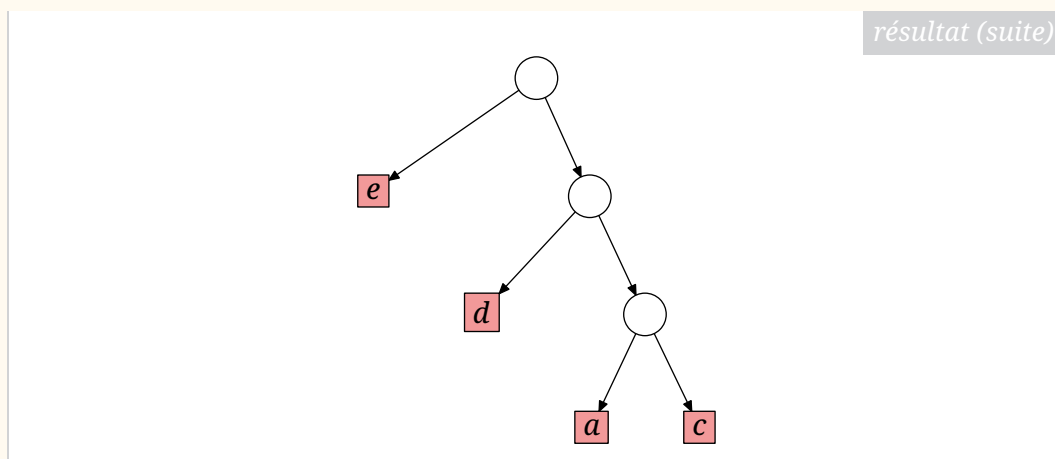
code

```

1 verbatimtex \leavevmode etex;
2 input huffman;
3 show_bits:=false;
4 show_node_values:=false;
5 show_leaf_values:=false;
6 set_node_size(8pt);
7 string charList[];
8 numeric weight[];
9 beginfig(1);
10 charList[4]:="a"; weight[4]:=4;
11 charList[3]:="c"; weight[3]:=6;
12 charList[1]:="d"; weight[1]:=7;
13 charList[2]:="e"; weight[2]:=10;
14
15 newBinHuffman.myHuff(4)(charList,weight) "hbsep(1cm)";
16 myHuff.c=origin;
17 drawObj(myHuff);
18 endfig;

```





## Pour aller plus loin

Si les sujets de cet article vous ont intéressés, voici quelques références pour approfondir le sujet.

**Sur l’algorithme de Huffman** : on pourra consulter la page Wikipédia dédiée<sup>40</sup>. On pourra aussi lire en anglais la publication de David A. Huffman de 1951<sup>41</sup>, ou [2, section 2.3] de Donald E. KNUTH.

**Sur METAPOST** : on pourra consulter les publications dans les *Cahiers GUTenberg*<sup>42</sup>, notamment [3] de Fabrice POPINEAU, [4] de Yves SOULET ou enfin [5] de Denis ROEGEL. Pour voir METAPOST en action, on pourra aussi consulter le site web de l’association GUTenberg : <https://metapost.gutenberg-asso.fr>.

**Sur METAOBJ** : on pourra consulter sa très riche documentation<sup>43</sup> ainsi que la section dédiée de *The LaTeX Graphics Companion*.

## Références

- [1] Michel GOOSSENS et al. *The LaTeX Graphics Companion*. 2e édition. Upper Saddle River, NJ : Addison Wesley, 2 août 2007. 976 p. ISBN : 978-0-321-50892-8.
- [2] Donald E. KNUTH. *The Art of Computer Programming : Volume 1 : Fundamental Algorithms*. 3e édition. Reading, Mass : Addison Wesley, 7 juill. 1997. 672 p. ISBN : 978-0-201-89683-1.
- [3] Fabrice POPINEAU. « MetaPost pratique ». In : *Cahiers GUTenberg* 41 (2001), p. 167-175. URL : [http://www.numdam.org/item/CG\\_2001\\_\\_41\\_167\\_0/](http://www.numdam.org/item/CG_2001__41_167_0/).
- [4] Yves SOULET. « MetaPost raconté aux piétons ». In : *Cahiers GUTenberg* 52-53 (2009), p. 5-117. URL : [http://www.numdam.org/item/CG\\_2009\\_\\_52-53\\_5\\_](http://www.numdam.org/item/CG_2009__52-53_5_).
- [5] Denis ROEGEL. « MetaPost l’intelligence graphique ». In : *Cahiers GUTenberg* 41 (2001), p. 5-16. URL : [http://www.numdam.org/item/CG\\_2001\\_\\_41\\_5\\_0/](http://www.numdam.org/item/CG_2001__41_5_0/).



Maxime Chupin

40. [https://fr.wikipedia.org/wiki/Codage\\_de\\_Huffman](https://fr.wikipedia.org/wiki/Codage_de_Huffman)

41. <https://www.ias.ac.in/article/fulltext/reso/011/02/0091-0099>

42. <http://www.numdam.org/search/Metapost-q&eprint%3DFalse/>

43. <https://ctan.org/pkg/metaobj>

## 🌀 DANS LE LABYRINTHE INFERNAL DES ENVIRONNEMENTS D'AMSMATH

### Un peu de contexte

Il y a quelques mois, j'ai dispensé une formation  $\LaTeX$  de niveau avancé à des doctorants à l'université de Lille. Inévitablement, il y a été question de composition des mathématiques et j'y ai présenté les plus courants des environnements du package `amsmath` permettant de composer des formules « hors-texte » : `equation`, `cases`, `multline` et `align`. Mais j'étais tombé cette fois-là sur un public plus coriace que d'habitude qui, connaissant au moins de nom les autres environnements de ce package, a voulu avoir des précisions à leur sujet. J'ai répondu de façon improvisée et, avouons-le, parfois confuse, n'utilisant pas moi-même tous ces environnements.

Voulant pallier cette déficience de mon cours, j'ai consulté la documentation du package `amsmath` qui ne m'a pas toujours paru d'une grande clarté. En cherchant sur Internet davantage d'informations, je suis tombé notamment sur le très [instructif et beau diagramme](#) de Stefan Kottwitz qui m'aurait permis de répondre à la plupart des questions posées par les doctorants. Mais, anticipant des futures demandes plus précises, j'ai quelque peu affiné ce diagramme (ce qui a nécessité que je me plonge cependant dans la documentation d'`amsmath`).

### Diagramme labyrinthique

Comme il est possible que ce diagramme puisse être utile à d'autres que ceux que je forme à  $\LaTeX$ , je le partage <sup>44</sup> ici, page 44; les noms d'environnements qui s'y trouvent (dans des boîtes sur fond rose) sont des liens hypertextes vers un exemple correspondant.

Ce diagramme et les exemples afférents devraient rendre clair l'usage de la plupart de ces environnements, à l'exception peut-être de ceux qui suivent la question « espaces `autom[atiques]` entre paires de colonnes ? » : `alignat`, `align` et `flalign`. Pour bien les comprendre, il n'est pas inutile de savoir que :

- ils créent des paires de colonnes alignées pour la première à droite et pour la deuxième à gauche, chacune de ces paires étant à séparer par `&`;
- `alignat` nécessite en outre que lui soit spécifié en argument obligatoire le nombre de paires de colonnes à composer.

44. Tous les diaporamas de mes cours  $\LaTeX$  sont cependant en accès libre à l'adresse <https://dgxy.link/cours-latex>.

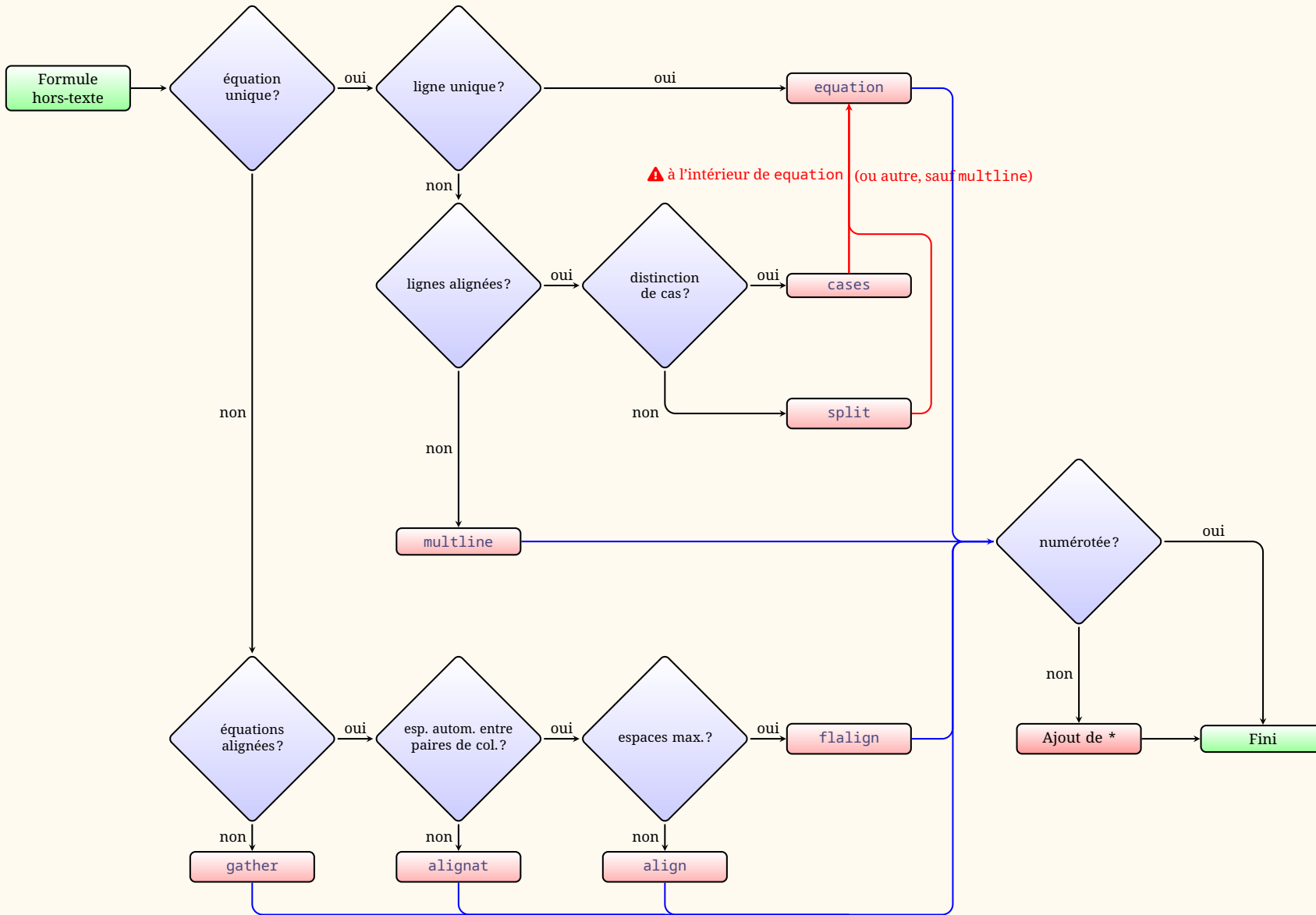


FIGURE 2 – Jojo est perdu dans le labyrinthe des environnements du package `amsmath`. Sauras-tu l'aider à choisir celui qui convient pour qu'il finisse sa formule hors-texte ?

## Exemples

### Exemple 34 : Éq. unique, ligne unique

```

1 \begin{equation}
2   a=b
3 \end{equation}

```

code

résultat

$$a = b \quad (1)$$

### Exemple 35 : Éq. unique, lignes multiples alignées (distinction de cas)

```

1 \begin{equation}
2   |a| =
3   \begin{cases}
4     a & \text{si } a \geq 0 \\
5     -a & \text{si } a < 0
6   \end{cases}
7 \end{equation}

```

code

résultat

$$|a| = \begin{cases} a & \text{si } a \geq 0 \\ -a & \text{si } a < 0 \end{cases} \quad (2)$$

### Exemple 36 : Éq. unique, lignes multiples alignées

```

1 \begin{equation}
2   \begin{split}
3     a & = b \\
4     & = c
5   \end{split}
6 \end{equation}

```

code

résultat

$$\begin{aligned} a &= b \\ &= c \end{aligned} \quad (3)$$

### Exemple 37 : Éq. unique, lignes multiples non alignées

```

1 \begin{multline}
2   a+b+c+d+e+f+g+h+i+j+k+l+m \\
3   =n+o+p+q+r+s+t+u

```

code

4	<code>+v+w+x+y+z</code>	code (suite)
5	<code>\end{multline}</code>	
<div style="text-align: right; font-size: small; color: #808080; margin-bottom: 5px;">résultat</div> $  \begin{aligned}  & a+b+c+d+e+f+g+h+i+j+k+l+m \\  & \qquad \qquad \qquad =n+o+p+q+r+s+t+u \\  & \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad +v+w+x+y+z \quad (4)  \end{aligned}  $		

## Exemple 38 : Éq. multiples non alignées

1	<code>\begin{gather}</code>	code
2	<code>  a=b+c \\\</code>	
3	<code>  c+d=e</code>	
4	<code>\end{gather}</code>	
<div style="text-align: right; font-size: small; color: #808080; margin-bottom: 5px;">résultat</div> $  \begin{aligned}  a &= b + c & (5) \\  c + d &= e & (6)  \end{aligned}  $		

## Exemple 39 : Éq. multiples alignées, sans esp. autom. entre paires de colonnes

1	<code>\begin{description}</code>	code
2	<code>\item[Effet généralement non souhaité :]</code>	
3	<code>  \begin{alignat}{2}</code>	
4	<code>    a_1 &amp; = b_1 &amp; &amp; c_1 &amp; = d_1 &amp; \\\</code>	
5	<code>    a_2 &amp; = b_2 &amp; &amp; c_2 &amp; = d_2+e_2</code>	
6	<code>  \end{alignat}</code>	
7	<code>\item[Effet généralement préféré :]</code>	
8	<code>  \begin{alignat}{2}</code>	
9	<code>    a_1 &amp; = b_1 &amp; &amp; \quad c_1 &amp; = d_1 &amp; \\\</code>	
10	<code>    a_2 &amp; = b_2 &amp; &amp; \quad c_2 &amp; = d_2+e_2</code>	
11	<code>  \end{alignat}</code>	
12	<code>\item[Effet généralement cherché :]</code>	
13	<code>  \begin{alignat}{2}</code>	
14	<code>    a_1 &amp; = b_1 &amp; &amp; \quad &amp; \text{Wow !} &amp; \\\</code>	
15	<code>    a_2 &amp; = b_2 &amp; &amp; \quad &amp; \text{Dingue !}</code>	
16	<code>  \end{alignat}</code>	
17	<code>\end{description}</code>	



## 🎵 LA FONTE DE CE NUMÉRO : NOTO

La présente *Lettre* est composée avec Noto... mais cette information n'est guère pertinente ! En effet, cette famille de fontes est très vaste. Nous vous renvoyons à l'article de Steven Matteson en page 53, qui est consacré au développement de ces fontes. Si les différentes pages de ce numéro vous ont permis d'apprécier le rendu de ces fontes, il est temps d'examiner avec précision leur utilisation avec  $\LaTeX$ .

### Petite illustration des caractères latins

Comme nous le faisons régulièrement, nous présentons en figure 3 des échantillons montrant la fonte *Noto Serif*. De la même manière, les figures 4 et 5 page suivante illustrent respectivement les fontes *Noto Sans* et *Noto Sans Mono*.

FIGURE 3 – La fonte *Noto Serif* : caractères romains, italiques et gras.



### Avec $\LaTeX$

Avec  $\LaTeX$  et plus encore avec les moteurs  $\text{Lua}\TeX$  et  $\text{Xe}\TeX$ , qui permettent d'utiliser le package `fontspec`, composer des documents à l'aide des familles de fonte *Noto* est aisé. Examinons les outils à notre disposition.

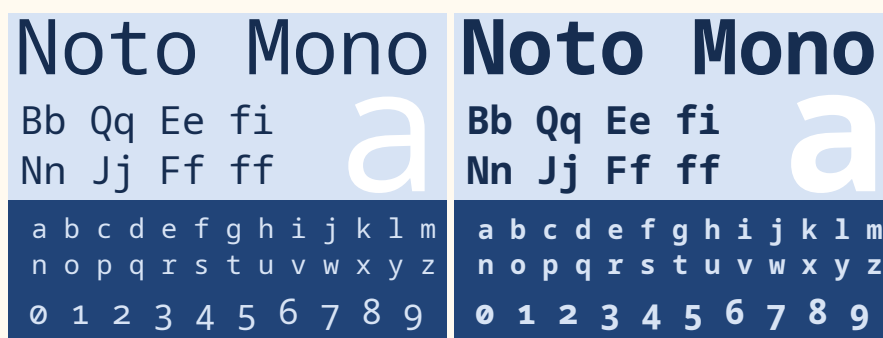
### Le package pour les caractères latins

Pour les caractères latins, Bob Tennent fournit le package `noto` qui fonctionne avec les moteurs  $\LaTeX$ ,  $\text{pdf}\LaTeX$ ,  $\text{Xe}\LaTeX$  et  $\text{Lua}\LaTeX$ . Plus précisément, ce package permet d'utiliser les fontes *Noto Serif*, *Noto Sans* et *Noto Sans Mono* présentées précédemment. Comme le dit Steven Matteson dans son article (voir page 66), « les polices Noto sont construites comme des modules individuels couvrant une ou quelques gammes de scripts Unicode, ou systèmes d'écriture. » Il est donc logique que des packages  $\LaTeX$  reflètent cet ordonnancement.

FIGURE 4 – La fonte *Noto Sans* : caractères romains, italiques et gras.



FIGURE 5 – Les glyphes courants et gras de la fonte *Noto Sans Mono*.



Le fichier README du package `noto` nous indique que<sup>45</sup> :

**le package `noto` :**

- charge *Noto Serif* comme fonte romane par défaut;
- charge *Noto Sans* comme fonte linéale par défaut;
- charge *Noto Sans Mono* comme fonte monospace par défaut;
- définit `\familydefault` comme `\rmdefault`;
- l'option `scale` n'affecte pas *Noto Serif*.

**Le package `noto-serif` :**

- charge *Noto Serif* comme fonte romane par défaut;
- définit `\familydefault` comme `\rmdefault`;
- ni *Noto Sans*, ni *Noto Sans Mono* ne sont chargées;
- l'option `scale` est utilisable.

**Le package `noto-sans` :**

- charge *Noto Sans* comme fonte linéale par défaut;
- ne modifie pas `\familydefault`;

45. Description traduite de la documentation de `notomath`.



- ni *Noto Serif*, ni *Noto Sans Mono* ne sont chargées;
- l’option `scale` est utilisable;
- définit `\familydefault` comme `\sfdefault` si l’option `sfdefault` est employée.

#### Le package `noto-mono` :

- charge *Noto Sans Mono* comme fonte monospace par défaut;
- ni *Noto Serif*, ni *Noto Sans* ne sont chargées;
- l’option `scale` est utilisable.

### Les autres packages

D’autres packages sont accessibles sur le CTAN. Il y a par exemple les équivalents des packages précédemment cités pour les versions condensées des fontes, qui sont à retrouver sur le CTAN sous le nom de `notocondensed`. Vous pouvez voir une illustration de cette famille (en serif seulement) en figure 6. Ce package n’est pas automatiquement délivré avec la T<sub>E</sub>X Live, mais fait partie de la T<sub>E</sub>X Live *contrib*, et il vous faudra suivre les indications données ici : <https://contrib.texlive.info> pour l’installer chez vous.

FIGURE 6 – La fonte *Noto Condensed* : caractères romains, italiques et gras.



Les packages `noto-emoji` et `notoCJKsc` fournissent<sup>46</sup> respectivement les fontes en TrueType et OpenType relatives aux émoticônes et aux familles serifs et sans-serifs des fontes pour le chinois traditionnel et simplifié, le japonais et le coréen.

Ainsi pour composer un texte en japonais avec la fonte *Noto*, on pourra écrire, en utilisant Lua<sub>T</sub>E<sub>X</sub> ou X<sub>T</sub><sub>L</sub>A<sub>T</sub>E<sub>X</sub> :

46. Et ne semblent faire que cela.

## Exemple 42

```
\Large\fontspec{NotoSansCJK-Regular}
日本語
```

code

日本語

résultat

De la même manière, le package `noto-emoji` permet d'utiliser la fonte *Noto Emoji* qui fournit des émoticônes comme le montre l'exemple suivant. Pour utiliser cette fonte, on charge le package `emoji`<sup>47</sup> à l'aide de

## Exemple 43

```
1 \usepackage{emoji}
```

ce qui permet de faire simplement appel aux émoticônes avec la commande `\emoji`.

## Exemple 44

```
1 \setemojifont{Noto Color Emoji}
2 \emoji{duck}\emoji{+1}\emoji{joy}
```

code



résultat

## Avec fontspec

Dès lors que l'on utilise Lua $\text{\LaTeX}$  ou Xe $\text{\LaTeX}$ , on a accès au package `fontspec` qui permet d'utiliser très simplement de nombreuses fontes.

Il suffira de télécharger la fonte *Noto* que l'on souhaite utiliser, l'installer (sur son système ou la mettre dans son répertoire de travail) et y faire appel avec `fontspec`.

Illustrons ceci avec une fonte pour l'arabe<sup>48</sup>. Nous téléchargeons donc la fonte *Noto Naskh Arabic* sur le site <https://fonts.google.com><sup>49</sup>.

## Exemple 45

```
\Huge\textdir TRT\fontspec[Script=Arabic]{
  NotoNaskhArabic}
أحدث العربية في باريس
```

code

47. Le package `emoji` ne fonctionne qu'avec Lua $\text{\LaTeX}$ .

48. Pour l'arabe, on peut utiliser aussi le package `arabluatex` du francophone Robert Alessi. Il faudra définir la fonte avant de charger le package `arabluatex` comme indiqué dans la documentation.

49. Le projet Noto est hébergé sur le site <https://notofonts.github.io/>. Une simple visite sur cette page permet de constater l'étendue de ce projet.

résultat (suite)

## أتحدث العربية في باريس

L'utilisation du package `fontspec` permet donc, avec les moteurs Lua $\TeX$  et Xe $\TeX$ , d'utiliser toutes les fontes de la grande, très grande famille *Noto*.

### Composer des maths

Sur le CTAN, on dispose du package `notomath`, de Michael Sharpe, qui fournit les fontes *Noto Math* et *Noto Sans Math* pour composer les mathématiques avec les familles de fontes *Noto* pour les caractères latins. Le package va même au-delà : il paramètre, suivant les options que l'on lui donne, les différents packages `noto-serif`, `noto-sans` et `noto-mono`. Le package `notomath` utilise `newtxmath`, développé par le même auteur, pour configurer tous les détails de la fonte mathématique. Nous n'allons pas ici décrire toutes les options de ce package et nous vous renvoyons à sa documentation, qui est très complète. `notomath` passe même des options à `newtxmath` lors de son chargement ! Il sert d'interface pour simplifier l'alignement du texte et des mathématiques.

Par exemple pour configurer les fontes *Noto* avec le serif, le sans, le mono et les maths, on pourra utiliser le code suivant<sup>50</sup> :

#### Exemple 46

```
1 \usepackage[mono]{notomath}
2 % charge NotoSerif, NotoSans, NotoSansMono,
3 % et fixe mainfont=NotoSerif
```

En figure 7, on pourra observer un exemple concret (car les numéros de la *Lettre* manquent tragiquement de mathématiques !). On pourra voir en figure 8 le même exemple en sans serif.

FIGURE 7 – Illustration de la composition des mathématiques avec *Noto*.

**Theorème 1 (des résidus).** Soit  $f$  une fonction analytique dans une région  $G$  à l'exception des points isolés  $a_1, a_2, \dots, a_m$ . Si  $\gamma$  est une courbe dans  $G$  fermée simple, rectifiable, positivement orientée qui ne passe par aucun des points  $a_k$ , alors

$$\sum_k \operatorname{Res}(f, a_k) = \frac{1}{2\pi i} \int_{\gamma} f(z) dz.$$

FIGURE 8 – Illustration de la composition des mathématiques avec *Noto Sans*.

**Theorème 1 (des résidus).** Soit  $f$  une fonction analytique dans une région  $G$  à l'exception des points isolés  $a_1, a_2, \dots, a_m$ . Si  $\gamma$  est une courbe dans  $G$  fermée simple, rectifiable, positivement orientée qui ne passe par aucun des points  $a_k$ , alors

$$\sum_k \operatorname{Res}(f, a_k) = \frac{1}{2\pi i} \int_{\gamma} f(z) dz.$$

Malheureusement, en utilisant la fonte mathématique serif, le réglage par défaut du facteur d'échelle du package `notomath` ne nous semble pas très bon. Pour la figure 7, nous avons corrigé ce défaut en utilisant le code suivant :

50. Un *bug* semble se produire sur la fonte mono ; nous l'avons signalé à l'auteur.

## Exemple 47

```
1 \usepackage[scale=1.1]{notomath}
```

De même, pour harmoniser la hauteur d'x entre les fontes serif et sans-serif lors de l'utilisation de `notomath`, nous avons chargé les packages `noto` et `newtxmath` comme suit :

## Exemple 48

```
1 \usepackage{noto}
2 \usepackage[noto, scale=1.1]{newtxmath}
```

... et c'est ainsi que nous avons composé la présente *Lettre*!



Maxime Chupin

# CAHIERS GUTenberg

Il va bientôt sortir... le Cahier 59!

On y traitera de droit, de philologie,  
d'édition de textes classiques  
et... de traitements de texte `WYSIWYG`!

Mais tout en  $\LaTeX$ , bien sûr!

Ce numéro sera prochainement lancé en souscription.



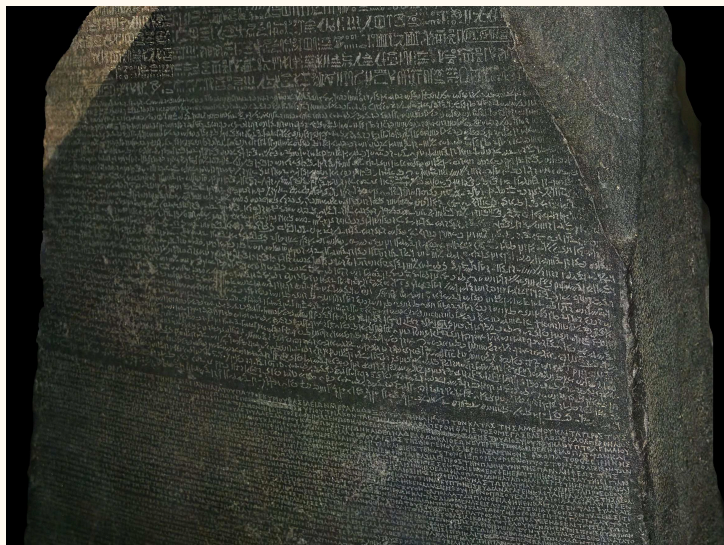
## 🌀 EN ROUTE VERS NOTO

Note de la rédaction du *TUGboat* : ce texte est la transcription de l'exposé de Steven Matteson lors de la conférence *TUG 2020*. Elle n'a fait l'objet que de retouches mineures. Certaines de ses illustrations sont ici omises. Leur intégralité, ainsi qu'une vidéo de la conférence, sont disponibles en ligne à l'adresse de la conférence : <https://tug.org/tug2020>.

La famille de polices Noto est l'une des plus grandes entreprises de l'histoire de la création de caractères. Elle n'alla certainement pas en ligne droite, d'un point A à un point B. J'ai été impliqué de façon intermittente pendant 14 ans, et environ 60 autres personnes y ont contribué jusqu'à ce jour. Cela n'inclut pas les efforts déployés pour les polices chinoises, japonaises et coréennes, dont les gens d'Adobe pourraient vous parler.

Pour les besoins de cet exposé, la route qui nous mène vers Noto commence par la pierre de Rosette (figure 9) : un fragment de tablette large d'environ un mètre, semblable au granit, avec un travail remarquable de la main et du ciseau, épelant soigneusement un décret impérial dans trois systèmes d'écriture différents.

FIGURE 9 – La pierre de Rosette, 196 avant notre ère

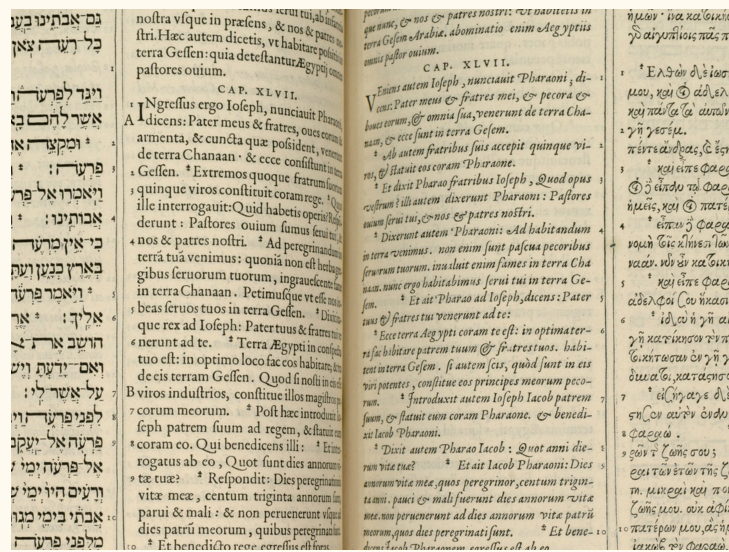


L'importance historique de cette pierre est légendaire. Je me souviens très bien de l'avoir étudiée en cours d'histoire à l'âge de 12-13 ans. Nous devions fabriquer notre propre version en pâte à modeler et y inscrire nos propres messages à l'aide d'un cure-dent. Nous avons même organisé un concours pour essayer de traduire les messages des autres. Merci aux écoles publiques d'Oak Park, dans l'Illinois !

Il s'agit du « décret de Memphis » du roi Ptolémée V, promulgué en 196 avant notre ère, au cours d'une période de turbulences politiques et de bouleversements culturels. La pierre de Rosette est un exemple fascinant des efforts laborieux déployés pour produire un document en plusieurs langues — en l'occurrence en hiéroglyphes, en écriture démotique et en grec ancien — dans le but de permettre à plusieurs cultures et générations de comprendre ce message unique.

Avance rapide jusqu'en 1573, où la Bible polyglotte de Christophe Plantin, financée par le roi Philippe II d'Espagne, est imprimée à Anvers (figure 10).

FIGURE 10 – La Bible polyglotte de Christophe Plantin, 1573



Les Écritures y sont imprimées en six volumes ; un dictionnaire multilingue en deux volumes les accompagnaient. Le texte est traduit en hébreu, en grec, en araméen, en syriaque et en latin. La typographie est stupéfiante de beauté, de simplicité et de minutie.

Le détail d'une double page présenté ci-dessus montre l'écriture hébraïque à l'extrême gauche et le latin dans une police de caractères romaine droite. À l'extrême droite se trouve un beau texte grec d'aspect cursif et fluide ; le latin qui l'accompagne est en italique pour compléter l'aspect et la texture du grec. Ce détail typographique contribue à unifier et à harmoniser la page malgré les différences entre les systèmes d'écriture multilingues.

Pour moi, la complexité du formatage et la qualité de l'impression sont d'une beauté inconcevable.

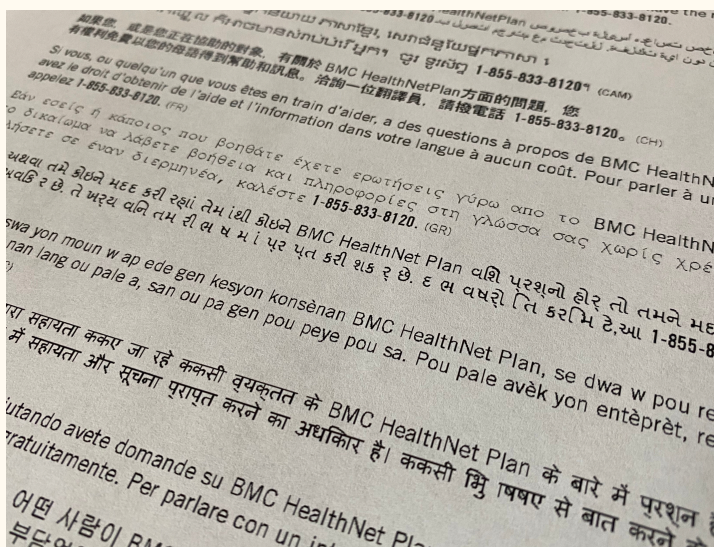
À partir de 1654, environ 70 ans plus tard, en Angleterre, l'évêque Brian Walton a commencé à travailler sur sa bible polyglotte (figure 11 page ci-contre). Il publie neuf traductions : araméen, hébreu, syriaque, arabe, samaritain, éthiopien, grec et latin. Cette production est financée par souscription et non par une subvention d'un gouvernement ou d'un organisme ecclésiastique. En l'espace d'un an, l'évêque a trouvé 400 contributeurs privés qui attendaient le livre fini.

Certains la considèrent comme la moins belle de toutes les bibles polyglottes. Cela peut s'expliquer par le fait qu'il ne s'agissait pas d'un projet financé par la royauté et assorti d'un faste royal proportionnel. Ou peut-être à cause de la complexité des neuf traductions par rapport aux quatre ou cinq traductions des ouvrages précédents, ce qui met le typographe au défi d'assembler une page harmonisée.

FIGURE 11 – La bible polyglotte de l'évêque Brian Walton, 1654



FIGURE 12 – Une page polyglotte contemporaine courante



Mais pour moi, la réalisation de Walton, la complexité du formatage et la qualité de l'impression sont d'une beauté inconcevable. Surtout si l'on la compare avec la situation dans laquelle nous nous trouvons 400 ans plus tard : il n'est qu'à voir ce texte détaillant les prestations d'une compagnie d'assurance quelconque (figure 12).

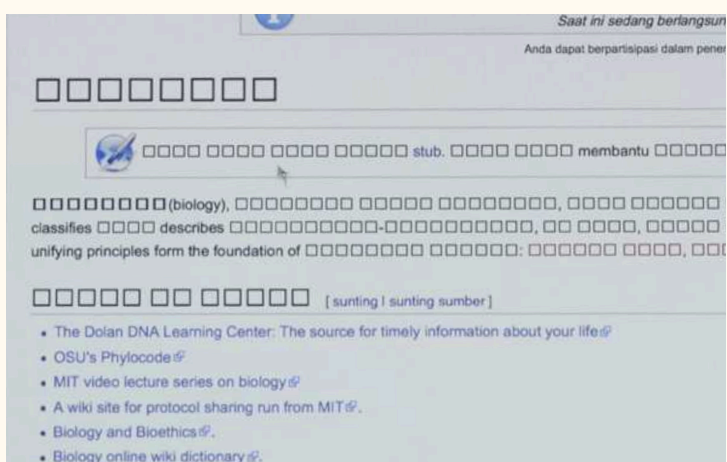
Cette page polyglotte est aujourd'hui courante et facile à réaliser avec nos outils de composition actuels. Cependant, comme de nombreux formulaires génériques ou institutionnels que nous voyons tous les jours, cette page peut être grandement améliorée. La longueur des lignes est excessive pour la plupart des textes représentés ici. La variété des styles de caractères donne l'impression qu'il s'agit d'une demande de rançon plutôt que d'un document sérieux. L'audace de certaines traductions les fait paraître beaucoup plus importantes que d'autres langues, ce qui n'est pas très égalitaire. Enfin, du point de vue de l'image de marque, l'identité visuelle de cette entreprise n'est pas respectée.

Je ne veux pas nier la complexité du travail ayant permis la création de cette

page. Elle dépasse largement les connaissances et les préoccupations du lecteur de cette police d'assurances. Le simple fait de pouvoir mettre en forme un texte de droite à gauche a constitué un grand pas en avant dans le domaine de l'informatique, sans parler des autres phénomènes mystérieux qui opèrent ici pour composer dans toutes ces langues.

Les pages typographiques multilingues, ou polyglottes, peuvent être pires que cela. Le pire cauchemar d'un auteur serait que son lecteur tombe sur une page qui ressemble à celle présentée en figure 13.

FIGURE 13 – Du tofu



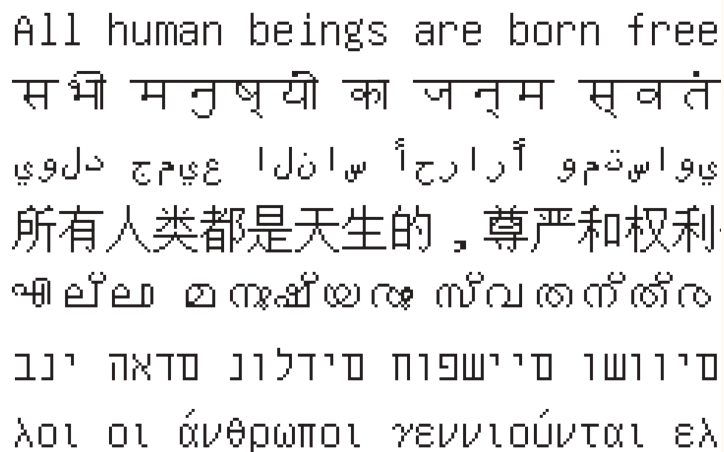
Cette page est pleine de glyphes manquants. Cette image m'a été transmise par Bob Jung, de Google, qui au sein de cette entreprise a orchestré une grande partie des débuts du projet Noto. S'il manque à un système informatique un caractère saisi par l'auteur, le lecteur voit le glyphe non défini, généralement un carré vide. Comme me l'a dit Bob, « ces carrés rappellent à beaucoup de gens, en particulier en Asie, les cubes de caillé de soja : le tofu. »

C'est de là que vient le nom des fontes Noto : pas du tofu, en anglais *No To(fu)* = Noto.

Mon voyage vers Noto a commencé en 1985, lorsque j'ai intégré l'école d'imprimerie de l'Institut de technologie de Rochester (*Rochester Institute of Technology*, ou RIT). Dans mes cours de typographie, j'ai été initié tant à la fonte chaude qu'à la fonte froide, cette dernière étant le tout dernier équipement de composition informatique de l'époque, c'est-à-dire les polices bitmap (figure 14 page suivante).



FIGURE 14 – Ma route vers Noto commence en 1985



Le professeur Archie Provan travaillait en tant que consultant pour Xerox dans le cadre de leur projet visant à rassembler des polices bitmap provenant de nombreuses fonderies différentes et capables de composer toutes les langues du monde. Il travaillait avec Ed Smura sur la norme AFII (*Association for Font Information Interchange*). Ce projet comprenait d'autres aspects liés à la typographie, notamment la classification des polices de caractères, mais le projet de polices bitmap était probablement l'aspect le plus ambitieux.

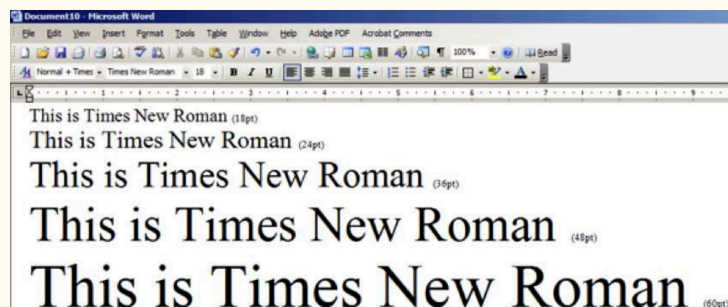
Grâce à ces polices bitmap, le système de publication Xerox Star, précurseur du Macintosh, devait pouvoir produire des documents dans n'importe quelle langue. Le rêve était que les polices soient transmises par satellite à un Xerox Star installé n'importe où dans le monde.

Lorsqu'il m'a décrit le projet, Archie m'a fait part d'une idée romantique selon laquelle, si les gens du monde entier pouvaient communiquer plus précisément et plus facilement, ils passeraient moins de temps à se battre.

Aussi banales soient-elles, ces formes bitmap monochromes allaient véhiculer la puissance de nos propres messages, de nos propres « décrets de Memphis », dans le monde entier.

Lorsque j'ai obtenu mon diplôme, les polices vectorielles que nous utilisons aujourd'hui se professionnalisaient. Les formes bitmap irrégulières de taille unique n'étaient soudain plus suffisantes, puisqu'il était désormais possible d'agrandir un caractère à l'infini pour obtenir une typographie plus expressive (figure 15).

FIGURE 15 – Premières polices de caractères vectorielles



J'ai commencé à travailler sur les polices du système TrueType pour Microsoft

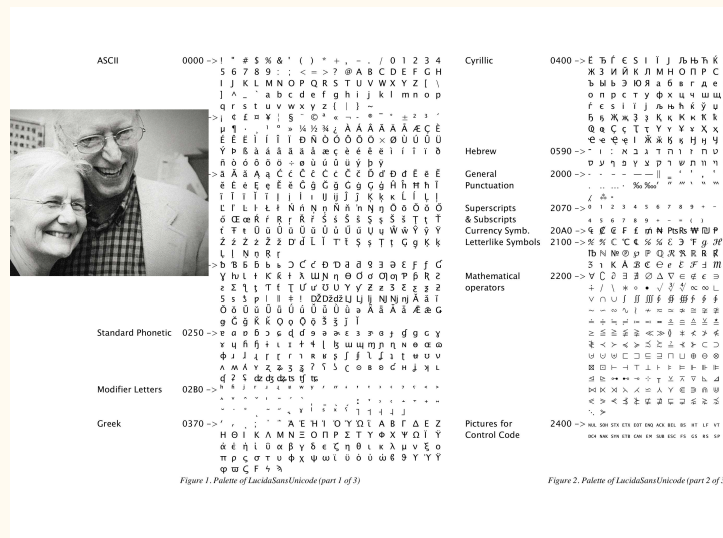
en 1990. Ces polices avaient toutes un jeu de caractères modeste d'environ 300 caractères, ce qui était déjà plus que ce que les premiers formats de polices pouvaient gérer. Les polices TrueType pouvaient gérer plus de caractères, et donc composer plus de langues, parce que l'Unicode était devenu le moyen standard de coder ou « d'organiser » toutes les lettres de tous les alphabets du monde, autorisant (à l'époque) 65 536 caractères au total. Les systèmes d'encodage précédents, y compris ISO 8859, n'autorisaient généralement que 256 caractères dans un seul fichier.

Avec Unicode, chaque caractère dans le monde reçoit un identifiant unique. Par exemple, le « G doux » du turc reçoit un nom et un numéro Unicode (Ğ, U+011E), tout comme l'omega grec (Ω, U+03A9), etc. Le registre Unicode est constamment mis à jour. En 1999, par exemple, le consortium Unicode s'est empressé de placer le symbole de l'euro (€, U+20AC) dans son répertoire afin que les fondries de polices de caractères comme Monotype puissent mettre à jour leurs énormes bibliothèques de polices pour prendre en charge la nouvelle monnaie.

Entre 1990 et 1993, à jet continu, les fondries ont continué à développer des polices codées en Unicode, dont la taille ne cessait de croître. Le jeu de caractères WGL (ou *Windows Glyph List*), défini par Microsoft, augmentait les attentes en matière de polices de caractères, qui devaient prendre en charge environ 90 langues, avec environ 600 caractères uniques dans un fichier de police.

Bigelow et Holmes ont créé leur police Lucida Sans Unicode pour coïncider avec la publication de la spécification 1.0 d'Unicode (figure 16). Lucida Sans Unicode a été publié par Microsoft en 1993 et a ajouté le grec, le cyrillique et l'hébreu au jeu de caractères latins. La police prenait également en charge les symboles phonétiques et mathématiques définis par l'Unicode pour la version 1.0.

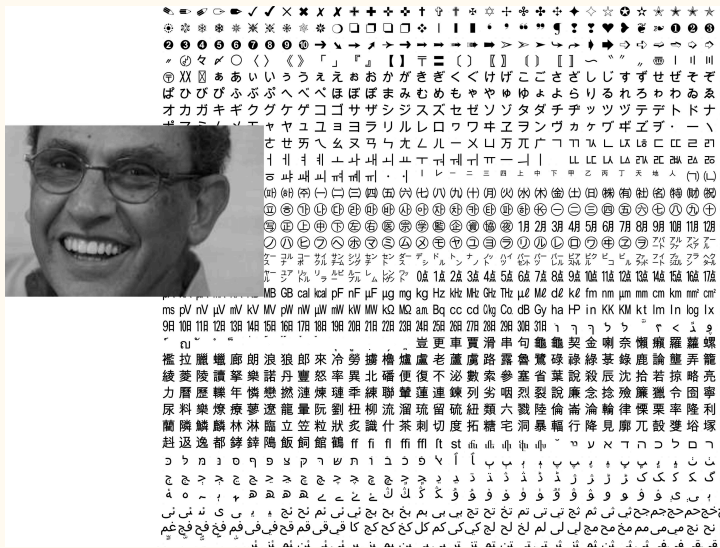
FIGURE 16 – Bigelow & Holmes Lucida Sans Unicode, 1993



En 1997, Monotype a été chargé par Microsoft d'étendre Arial à l'ensemble de l'Unicode 2.0, soit seulement 50 000 formes de lettres pour un poids de 22 Mo pour le seul fichier de police. Bien que le format de police TrueType puisse théoriquement prendre en charge tous ces caractères dans un seul fichier de police, il s'agissait d'un processus délicat à mettre en œuvre. Nous avons

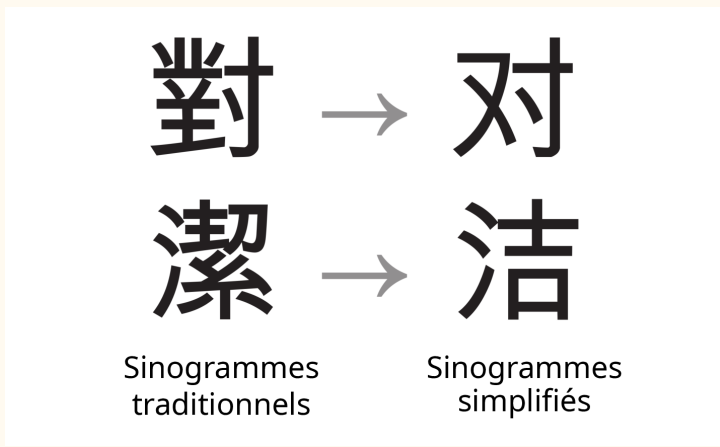
dû créer de nombreuses petites fontes (que nous appelions *fontlets*) avant de les assembler à la fin du processus. Mon collègue de longue date, Kamal Mansour, a veillé à ce que la conception soit, autant que possible pour l'époque, en harmonie avec Arial (figure 17).

FIGURE 17 – Monotype Arial pour Unicode 2.0, 1997



Nous avons alors inclus beaucoup plus d'idéogrammes chinois que ceux définis par Unicode. Les glyphes supplémentaires ont été inclus pour prendre en charge à la fois le chinois simplifié et le chinois traditionnel, de sorte que le nombre réel de formes de lettres dans la police dépasse celui de la norme Unicode (figure 18).

FIGURE 18 – Même Unicode, formes différentes



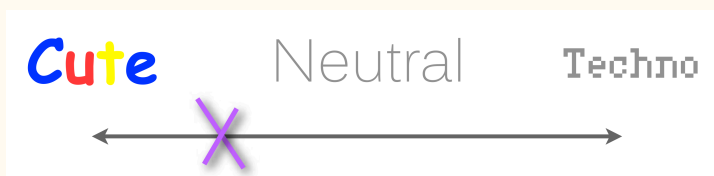
La figure 18 montre comment le même caractère Unicode peut être représenté par deux glyphes différents. Un équivalent occidental pourrait être de montrer la lettre *g* à la fois sous forme de boucle simple et de boucle double - il s'agit du même caractère Unicode représenté par deux formes de glyphes différents.

En 2006, Google m'a demandé de créer une famille de caractères pour une nouvelle plateforme de téléphonie mobile. Google voulait une interface utilisateur unique pour la marque Android. Une expérience d'interface unique commence, bien sûr, par le type de police de caractères avec lequel interagir.

En raison de l'image de marque quelque peu décalée de Google et d'Android,

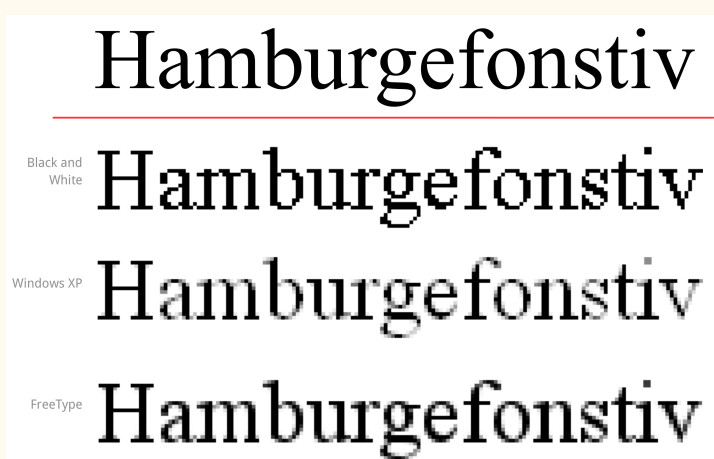
j'avais besoin de dessiner une police de caractères accessible qui le soit aussi; une fonte biaisant avec la neutralité : si elle était trop neutre, elle ne se démarquerait pas comme étant unique à la marque. Mais si elle était trop mignonne ou trop techno, la lisibilité et la fonctionnalité en pâtiraient (figure 19).

FIGURE 19 – Nouvel objectif de création de caractères pour Android



Android disposait également d'une interface graphique très spécifique. Rappelons que les écrans des téléphones portables étaient loin d'avoir la résolution des appareils d'aujourd'hui (figure 20).

FIGURE 20 – Différents écrans, différents résultats, 2006



L'exemple en figure 21 montre qu'un même dessin peut avoir un aspect très différent selon l'écran sur lequel il est visualisé. Les parties étroites des lettres peuvent sembler disparaître ou être légèrement trop lourdes, selon la manière dont le logiciel interprète et dessine les lettres. Cet exemple présente quelques dessins et expériences préliminaires qui furent testés.

FIGURE 21 – Premiers dessins pour la police Android

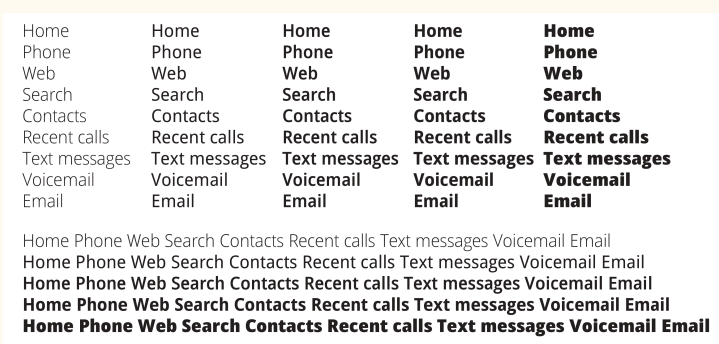


Nous avons dû créer des polices de caractères et les installer sur des appareils pour observer les effets de petits changements dans la conception et la proportion des lettres individuelles. Il s’agissait d’un processus laborieux car l’interface utilisateur était conçue en même temps que la police de caractères. Pour compliquer encore les choses, le matériel qui allait devenir les premiers téléphones Android était également en cours de conception et de fabrication.

J’ai conçu les polices en tenant compte de la façon dont elles s’afficheraient sur les différents écrans Android. J’ai travaillé avec l’équipe chargée de l’interface utilisateur pour m’assurer qu’il y avait suffisamment de contraste entre les graisses normales et les graisses grasses afin d’établir une hiérarchie dans l’interface. Je me suis assuré que les détails de chaque *design* fonctionnaient bien dans ces résolutions limitées.

Une fois que nous avons été sur la bonne voie avec le *design*, nous avons fait des tests de graisse pour voir quel contraste était nécessaire entre le texte courant et le gras (figure 22).

FIGURE 22 – Tests de graisse pour la police Android



Seules deux graisses étaient nécessaires ; les italiques seraient synthétisées par Android afin d’économiser de l’espace de stockage. D’autre part, après de nombreuses discussions, il a été décidé qu’une police avec empattements devait faire partie de l’ensemble des polices de base. Les polices avec empattements serviraient à la lecture des fils d’actualité et des textes longs. Les polices sans empattements seraient destinées aux éléments de l’interface utilisateur et aux menus.

FIGURE 23 – Famille de fontes Droid originale



Toutes les polices prenaient en charge le jeu de caractères WGL-4, mais on sentait qu’il allait être étendu si Android était un succès.

L’objectif était bien sûr de créer une famille de polices qui tienne la route sur des écrans de petite taille et qui donne à la plateforme une apparence accessible et conviviale. Nous y sommes peut-être parvenus, car un rédacteur de l’édition en ligne du magazine *Wired* a qualifié les polices de *googly*.

FIGURE 24 – Test d’affichage de Droid



Lorsque le moment est venu de décider de ce qui pouvait être affiché dans l’interface utilisateur d’Android, Google a estimé qu’il était clair qu’ils ne voulaient pas que cela ressemble à l’exemple de gauche présenté en figure 25, qui est malheureusement assez similaire à la déclaration de prestations d’assurance montrée plus haut (fig. 4). L’entreprise souhaitait plutôt une présentation harmonisée de la « marque Android » dans toutes les langues qu’elle prenait en charge. Tous les scripts devraient avoir un aspect contemporain, accessible et ergonomique, plus proche de l’exemple à droite. (Toutes nos excuses pour la faute de frappe dans le texte arabe).

FIGURE 25 – Scripts non harmonisés à gauche ; harmonisés à droite



Concevoir des formes cohérentes pour des fontes qui ne possèdent pas de sources historiques communes est quelque peu délicat et dans certains cas il faut adopter un compromis pour qu’elles affichent une grasse similaire. Par exemple, les pleins des caractères arabes sont sur les horizontales alors qu’ils se placent sur les fûts verticaux des alphabets latins. Cette particularité impose une énorme différence de réglage des graisses pour les équilibrer. L’harmonisation visuelle peut être obtenue en exportant des particularités graphiques des formes latines telles que la forme, la grasse de l’extrémité des empattements et des terminaisons ou l’équilibre entre les pleins et les déliés.

FIGURE 26 – Harmonisations entre l'arabe et le latin

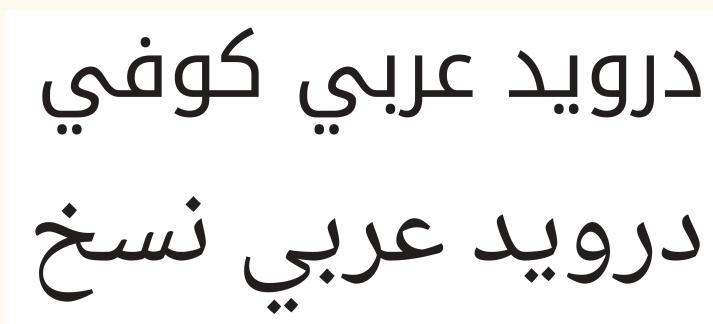


Les polices arabes ont été conçues par Pascal Zoghbi avec, de ma part, un peu de direction artistique. La graphie qui correspond le mieux aux caractères latins à empattements est appelée naskh. Dans la plupart des caractères naskh, les contreformes sont minuscules mais, dans ce cas, nous avons exagéré leur taille pour imiter celles des caractères latins et faciliter la lisibilité à l'écran.

Dans l'exemple figure 27, la ligne supérieure du texte arabe est composée dans un style appelé kufi (en français, il est parfois appelé coufi, ou coufique) qui se rapproche davantage d'une fonte latine sans empattements. Toutefois, les lecteurs arabes préfèrent généralement le style naskh (que l'on voit dans la ligne inférieure) pour une lecture prolongée. J'ai l'impression que cela ressemble à la résistance qu'avaient les lecteurs occidentaux à lire des livres écrits avec des caractères sans empattements. Ce n'est que dans les années 1950 et 1960 que les gens ont commencé à accepter cette nouvelle approche typographique.

En travaillant sur ce projet, j'ai appris que les lecteurs arabes étaient très habitués à devoir zoomer sur le texte dès qu'une page était chargée. L'un des objectifs de l'exagération des proportions était d'éviter d'avoir à zoomer pour lire les tailles arabes par défaut. D'après ce que j'ai compris de Google, nous y sommes parvenus avec Droid et Noto.

FIGURE 27 – Style kufi (en haut), style naskh (en bas)



Au-delà de l'arabe, il était urgent d'équiper Android de caractères thaïlandais. Il s'agit d'une fonte à empattements et l'on peut noter que certains détails font penser à la fonte latine du même type (l'exemple figure 28 page suivante et

les suivants sont tronqués à gauche et/ou à droite pour que les détails soient mieux visibles).

FIGURE 28 – Droid Thai, avec le latin pour comparaison



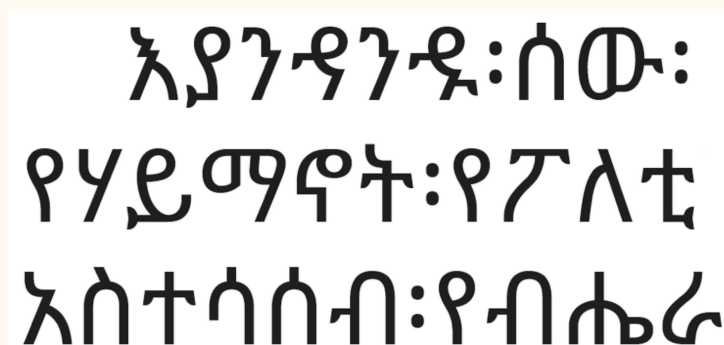
Un échantillon de la fonte hébraïque Droid Serif — une autre fonte où la difficulté réside dans la répartition des graisses, opposée à celle du latin.

FIGURE 29 – Droid hébreu



L'alphasyllabaire guèze est souvent représenté sous une forme inclinée, assez semblable à un italique. J'ai pensé qu'une forme droite serait plus lisible et plus utile pour l'interface utilisateur d'Android.

FIGURE 30 – Droid guèze



L'arménien s'inspire beaucoup des formes des minuscules latines, ce qui facilite considérablement l'harmonisation de ces caractères avec les lettres latines.



FIGURE 31 – Droid arménien

րդ էջերում: Պ  
 առշումն ընդու  
 պետությունն է  
 report հռչակս


De même, le géorgien s’inspire des formes rondes du latin. Sa texture est totalement différente de celle de l’arménien, mais il appartient clairement à la famille des polices de caractères Droid.

FIGURE 32 – Droid géorgien

1948 ვინაიდან ადამია  
 კრისათვის Australian  
 ანასწორი და განუყოფ  
 nyon აღიარება წარმო  
 მართლიანობის და va  
 ვინაიდან ადამიანის უ  
 გულებელყოფამ და აბ

Certains se demanderont peut-être pourquoi je parle d’Android à ce stade de mon voyage vers Noto.

FIGURE 33 – Les travaux se succèdent

	Droid Sans	Droid
	chrome	Open Sans
	Noto Sans	Noto

En 2009, on m’a demandé d’adapter Droid Sans pour en faire un caractère de marque pour Chrome. Ce dessin est devenu Open Sans, une version légèrement plus large du Droid, plus adaptée au texte normal dans les documents. Parallèlement, les équipes Chrome et Android de Google discutaient de l’idée d’unir leurs efforts pour développer une super police qui couvrirait l’ensemble de l’Unicode. Le problème de l’affichage de « tofu » dans les recherches sur internet devenait de plus en plus problématique à mesure que l’utilisation d’internet dans le monde augmentait de façon spectaculaire.

Avec l’arrivée conjointe de Chrome et d’Android, Open Sans est devenu la base de Noto Sans et Droid Serif est devenu Noto Serif.

En 2011, il a été décidé d’élargir la famille latine de Noto pour qu’elle comprenne une gamme typographique complète de familles, de graisses et de corps. Les versions condensées et étroites des caractères, avec et sans empattements, ont été ajoutées, ainsi que des graisses supplémentaires allant de l’ultra-léger à l’extra-noir. La fonte avec empattements a été enrichie au point d’offrir une gamme de gris typographiques allant du très dense au très clair. Cela représente environ 72 styles de police par famille — la palette typographique polyglotte n’est plus limitée aux seuls styles réguliers et gras !

FIGURE 34 – Noto, vers 2011

The image displays a collection of text samples from the Noto font family. It includes:
 

- Noto Sans** in a light weight.
- Noto Sans** in a bold weight.
- Noto Sans** in a regular weight.
- Noto Serif** in a regular weight.
- Noto Serif** in a bold weight.
- Noto Serif** in a regular weight.
- Noto Sans Monospaced** in a regular weight.

 The samples are arranged in a grid-like fashion, demonstrating the range of weights and styles available within the Noto family.

À cette époque, l’Unicode compte environ 3400 caractères pour prendre en charge l’écriture latine, cyrillique, grecque et phonétique. Cet ensemble de caractères permet de prendre en charge plus de 500 langues. Avec l’ajout de ces caractères supplémentaires, toutes les familles, graisses et corps des fontes latines supplémentaires signifiaient que l’on dessinait environ 230 000 caractères.

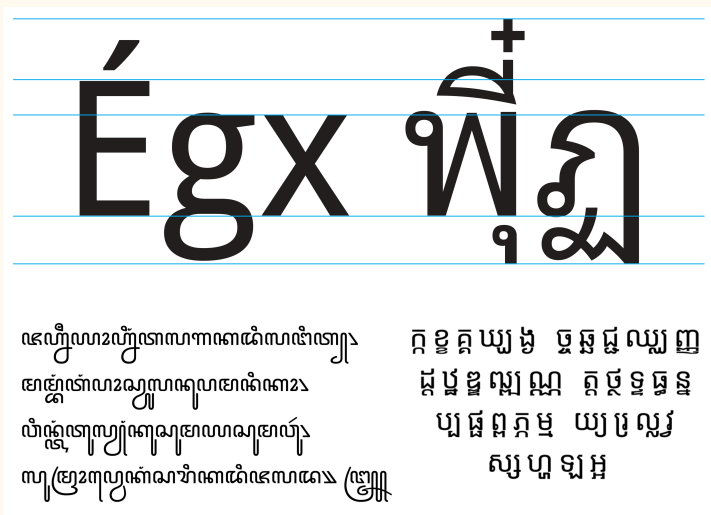
Il était désormais évident que Noto ne serait jamais livré sous la forme d’une seule police binaire avec l’ensemble de l’Unicode.

Rappelez-vous que l’ancienne police Unicode Arial pesait à elle seule 22 Mo et ne contenait que 55 000 caractères. Au lieu de cela, les polices Noto sont construites comme des modules individuels couvrant une ou quelques gammes de scripts Unicode, ou systèmes d’écriture.

Outre le problème de la taille des fichiers, certains systèmes d’écriture deviennent assez complexes dans leur forme, bien au-delà du thaï que nous avons dessiné. Verticalement, le thaï est tout juste capable de se glisser dans la métrique verticale d’une police de caractères latine. En figure 35, vous

pouvez voir que le javanais (à gauche) et le khmer (à droite) sont encore plus complexes verticalement, et ne peuvent tout simplement pas raisonnablement entrer dans les dimensions de l'écriture latine.

FIGURE 35 – Au-delà de la métrique latine : javanais (à gauche), khmer (à droite)



Un autre exemple est le style nastaliq de l'arabe utilisé pour l'ourdou, le pachtoune et le persan. Alors que le naskh, couramment utilisé, suit une ligne de base horizontale, la ligne de base du nastaliq est inclinée vers la gauche. Plus le mot est long, plus la ligne descend.

FIGURE 36 – Noto naskh (ligne de base plate) et Noto nastaliq (ligne de base inclinée)



L'une des façons dont nous avons envisagé cet énorme projet a été de diviser les systèmes d'écriture en catégories. De cette manière, nous pouvions classer les écritures apparentées en fonction de leur région, de leur complexité ou de leur utilisation relative dans le monde moderne.

Les cases oranges de la figure 37 page suivante indiquent des écritures qui peuvent être mortes depuis des centaines d'années ou plus, tandis que les vertes sont des écritures utilisées en Inde, et ainsi de suite. Cela a permis de

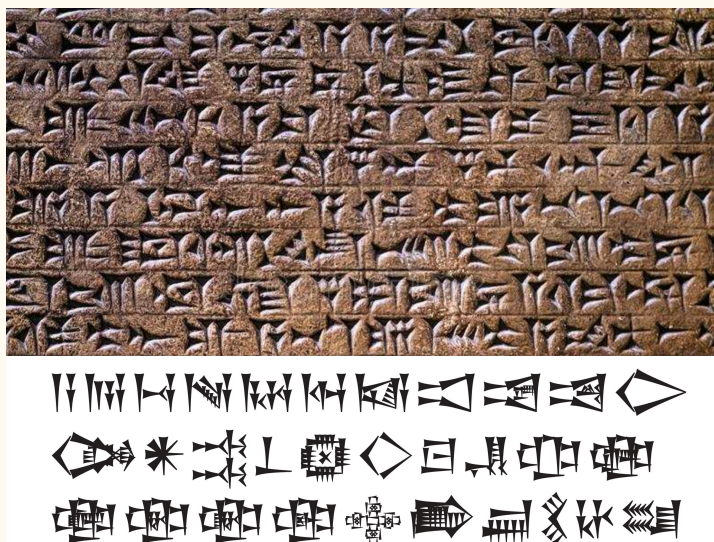
hiérarchiser et d’organiser l’expertise nécessaire à la réalisation de chaque élément du projet.

FIGURE 37 – Organisation des écritures pour Noto



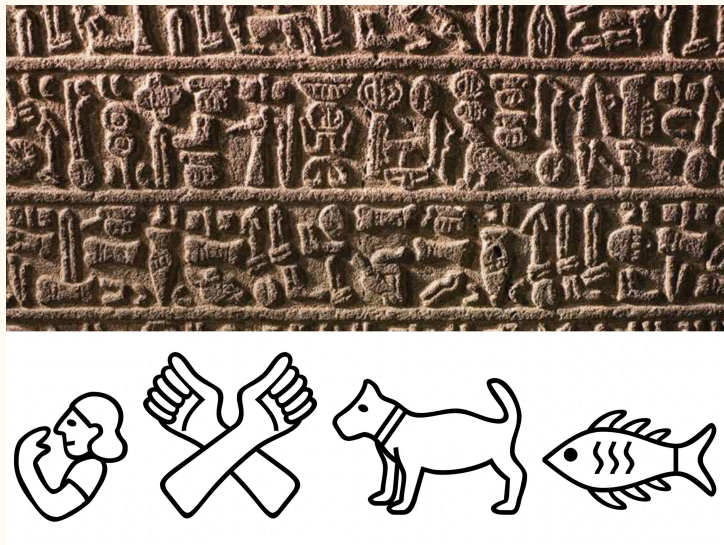
Les caractères cunéiformes sont un bon exemple des « écritures mortes » que nous venons de mentionner ; ils peuvent avoir jusqu’à 5000 ans. Bien qu’ils ne soient pas utilisés dans la pratique, ils sont certainement utiles pour les chercheurs et les linguistes de les avoir codés dans un fichier de polices de caractères.

FIGURE 38 – Cunéiforme, original et Noto



Les hiéroglyphes anatoliens datent d’au moins 4000 ans et sont donc antérieurs aux hiéroglyphes égyptiens. Ils sont donc représentés comme légèrement plus bruts et lissés.

FIGURE 39 – Hiéroglyphes anatoliens, originaux et Noto



Les hiéroglyphes égyptiens sont beaucoup plus nets et raffinés que leurs ancêtres anatoliens.

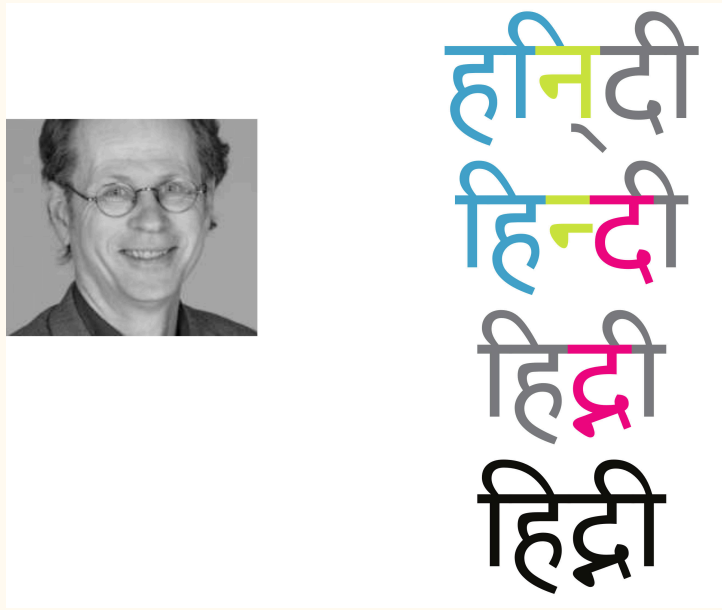
FIGURE 40 – Hiéroglyphes égyptiens, originaux et Noto



De nombreuses écritures sont classées comme « complexes » et nécessitent une grande quantité de programmation pour assembler les mots de la bonne manière. L'arabe, qui se lit de droite à gauche et comporte de nombreuses formes de la même lettre, entre dans cette catégorie, tout comme les écritures utilisées en Inde. Jelle Bosma est le responsable créatif de Monotype pour la création des écritures indiennes et travaille sur les mises à jour de la version 13 d'Unicode.

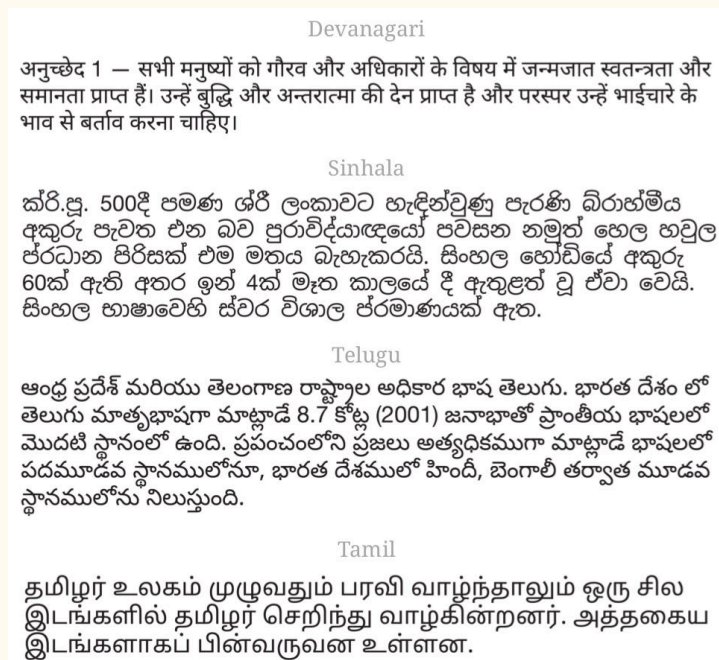
Dans l'illustration figure 41 page suivante, le mot de droite est le mot « Hindi » en écriture devanagari. La ligne supérieure montre l'aspect du mot avec des caractères Unicode simples assemblés pour épeler le mot. Chacune des lignes successives montre comment l'écriture est « remodelée » grâce à des tables typographiques avancées qui réorganisent les lettres au fur et à mesure qu'elles sont tapées. Sur la deuxième ligne, les caractères verts changent de place. Sur la troisième ligne, les caractères verts et roses forment une ligature.

FIGURE 41 – À gauche, le graphiste Jelle Bosma; à droite, le mot « Hindi » est formé au fur et à mesure de la frappe



Les systèmes d'écriture indiens semblent très différents les uns des autres, mais nous les avons conçus pour qu'ils s'harmonisent autant que possible. Le devanagari, assez rectangulaire, contraste beaucoup avec le cinghalais, qui est plus fluide, mais leur couleur et leur proportion sont préservées pour les maintenir en phase. Il en va de même pour le télougou et le tamoul, dont la texture générale est très différente, mais leurs proportions et leur couleur les relient.

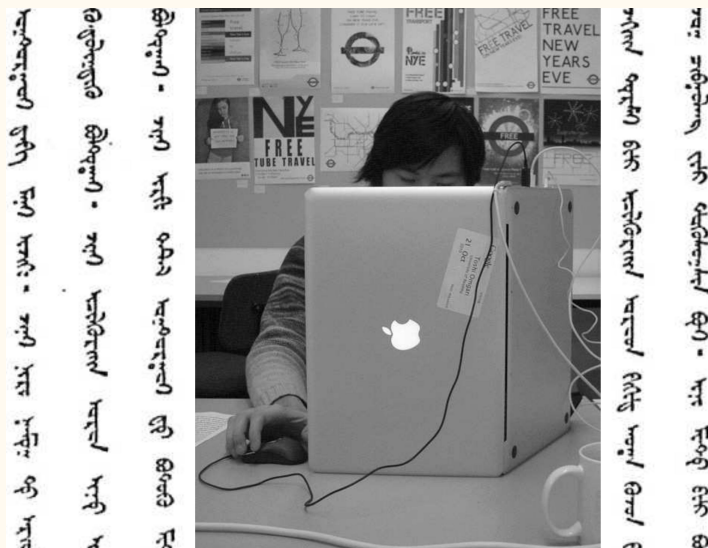
FIGURE 42 – Des écritures indiennes composées avec Noto



Tout comme j'ai mentionné qu'il y avait des problèmes techniques à résoudre avec l'Unicode Arial, il en va de même avec Noto. Mon collègue Toshi Omagari est représenté ci-dessous avec son ordinateur portable tourné de 90 degrés, ceci pour dessiner le mongol, une écriture connectée qui se lit de haut en bas, de gauche à droite.

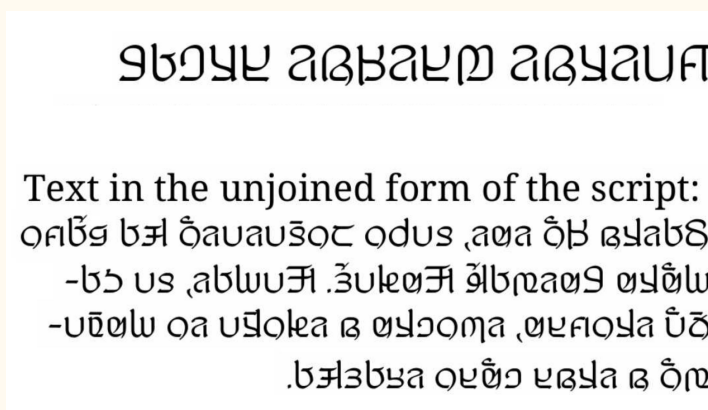
Les développeurs de l’outil GlyphsApp ont été d’un grand soutien en mettant à jour leur logiciel afin de rendre possible ces scripts complexes dans Noto. Il n’a pas fallu longtemps pour qu’ils fournissent une correction permettant à Toshi de voir son travail de la manière dont il serait utilisé.

FIGURE 43 – Toshi Omagari travaillant sur le mongol



J’ai parlé plus haut de « écritures mortes ». À l’opposé, il y a l’Adlam, une écriture développée à la fin des années 1980 par les frères Ibrahima et Abdoulaye Barry. Ce système d’écriture transcrit la langue peul parlée en Guinée, au Nigeria et au Liberia. Avant l’Adlam, le peul était écrit en arabe ou en latin.

FIGURE 44 – Adlam, développé dans les années 1980



En 2018, les polices Noto couvraient près de 64 000 caractères et de nombreuses mises à jour ont été effectuées au cours des deux dernières années. L’Unicode 13.0 ajoute 4 nouvelles écritures et 5000 nouveaux caractères à ce décompte. « Que reste-t-il encore ? » pourriez-vous vous demander.

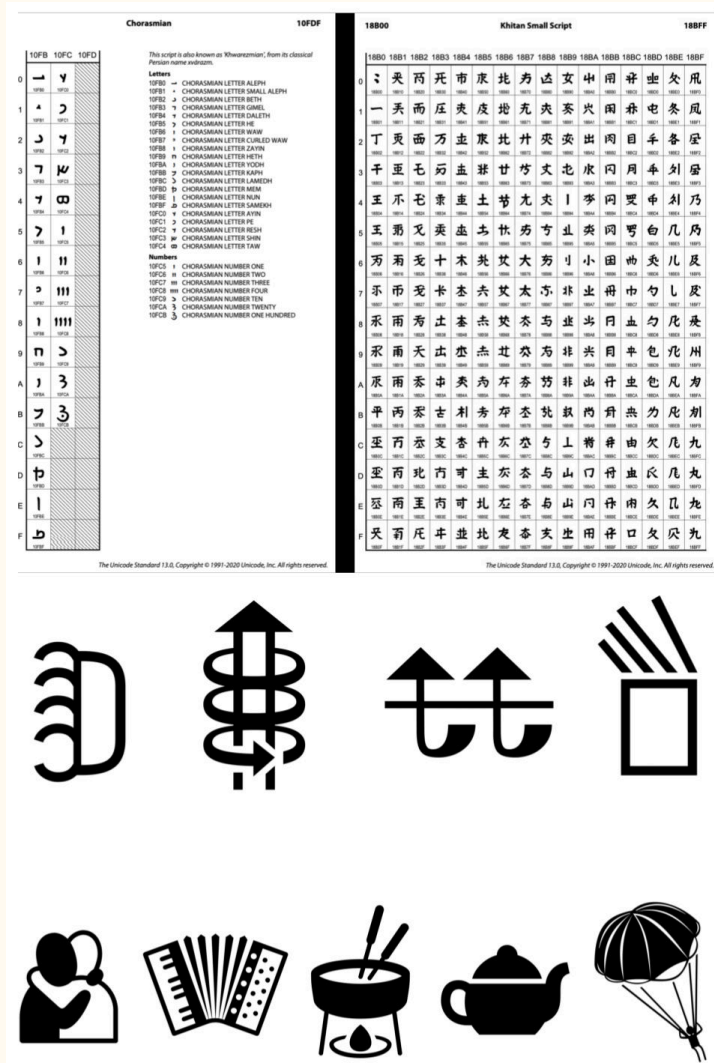
Le khitan, une langue autrefois parlée en Mandchourie, a été ajouté, de même que le chorasmien, une langue de l’ancienne Perse.

SignWriting<sup>51</sup>, un système de notation utilisé pour enseigner la langue des signes, a été ajouté. Il nécessite des milliers d’icônes pour représenter les gestes de la main et les expressions faciales utilisés par les interprètes en langue des signes.

51. <https://fr.wikipedia.org/wiki/SignWriting>

La police de symboles Noto s’enrichit de nombreux nouveaux caractères, dont les symboles tant attendus de l’accordéon et de la fondue <sup>52</sup>. La liste ne cesse de s’allonger.

FIGURE 45 – Ajoutés dans Unicode 13.0 : écritures khitan et chorasmienne, ou *khwarezmienne*, (en haut), *SignWriting* (au milieu) et différents symboles récents (en bas)



Noto continuera d’être améliorée pour refléter les demandes des différentes communautés humaines qui utilisent ces fontes.

Outre la trentaine de personnes de Monotype qui ont travaillé sur le projet Noto, nous avons collaboré avec plus de 30 linguistes, consultants et concepteurs extérieurs dans le monde entier, notamment Fiona Ross, Tiro Typeworks et Kigali Design.

J’aimerais également remercier d’autres personnes : Abdoulaye & Ibrahima Barry; Jo De Baerdemaeker; Cadson Demak Ltd; Diane Collier; Fontef Type Foundry; Kalapi Gajjar-Bordawekar; Yanone Gerner; Gajjar & Vilhjamsson Private Limited; Kimya Gandhi; Patrick Giasson; John Hudson; Indian Type Foundry; Yanek Iontef; Letterjuice Ltd.; Ben Mitchell; James Montalbano; Elena Papassissa; Rainer Erich Scheichelbauer; Zachary Scheuren; Georg Seifert; Vaibhav Singh; Anuthin Wongsunkakon; Pascal Zoghbi.

La communauté des utilisateurs et testeurs qui ont fourni un retour d’in-

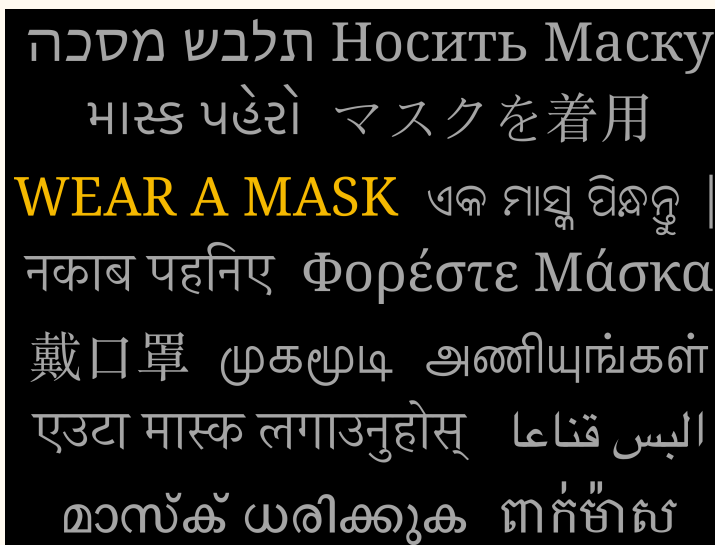
52. Quelle fondue, l’auteur ne le dit pas. La rédaction le regrette



formation est, bien entendu, beaucoup plus importante. C’est un honneur de travailler pour eux, de faire de cette énorme entreprise un outil à leur disposition.

En fin de compte, Noto ne sera peut-être pas utilisé pour retraduire la Bible ou des décrets impériaux. Il pourrait s’agir de messages très simples que nous pouvons transmettre avec cet énorme outil dont nous disposons. Et peut-être que Noto est un pas de plus vers la réalisation concrète de cette idée romantique que j’ai entendue lorsque j’étais étudiant, à savoir que les gens du monde entier pourraient passer plus de temps à communiquer plutôt qu’à se battre.

FIGURE 46 – Traductions non retouchées, via Google Translate



Steven Matteson

Traduction Patrick Bideault

Le traducteur remercie pour leur expertise Alain Joly et Claude Leloup, typographes.



## EN BREF

Des pourriels dans nos archives .....	74
TikZiT, un éditeur graphique pour TikZ.....	74
La séance solennelle de rentrée des cinq académies .....	75
Un appel à candidatures .....	75
<i>L<sup>A</sup>T<sub>E</sub>X Companion</i> , 3 <sup>e</sup> édition, 2023.....	76
(all)T <sub>E</sub> X au café .....	77
<i>Gustave</i> , un irrégulomadaire de poésie.....	78
Deux expositions.....	78
À la radio.....	79

### Des pourriels dans nos archives

Le travail administratif nous a récemment fait nous plonger dans les archives de nos listes de discussion. Et nous avons une nouvelle fois constaté qu'elles sont pleines de pourriels : certaines de nos adresses sont publiques, notamment celle du secrétariat, ce qui draine ces désagréables missives.

L'association dispose bien de différents outils de détection de ces indésirables, mais chaque jour quelques-uns réussissent à passer inaperçus. Ils sont donc distribués aux membres du bureau et archivés sur notre serveur.

Leur présence rend la consultation des archives de nos listes de diffusion pénible : ils représentent, selon les mois, entre la moitié et des trois-quarts des courriels archivés !

Nous avons donc commencé à faire le ménage ; nous supprimons quotidiennement ceux du jour et nettoiyons peu à peu les archives ; à l'heure actuelle, la liste de diffusion de la trésorerie, créée en 2012, a été entièrement nettoyée. Et les archives de la liste du secrétariat, qui reçoit bien plus de pourriels, ont été partiellement nettoyées : pour supprimer un message, trois clics sont nécessaires... et des milliers de messages inutiles ont été accumulés dans nos archives au fil des années !

Si parmi les adhérents se trouve un expert hyperkitty/mailman (ce sont les logiciels qui gèrent la liste et ses archives), qu'il n'hésite pas à se manifester : nous serions très heureux de disposer d'une interface plus pratique pour mener à bien cette nécessaire tâche. Les archives du secrétariat remontent à 2010, et dès cette date des pourriels y furent adressés, et donc archivés. Outre le coût de stockage des ces inutiles messages, nous aimerions laisser à nos successeurs des archives saines et utilisables. Par ailleurs, le paramétrage de spamassassin est complexe : nous serions heureux de bénéficier de l'expérience d'un expert pour affiner notre filtration. Avis aux bonnes volontés !

Patrick Bideault

### TikZiT, un éditeur graphique pour TikZ

J'ai découvert récemment le logiciel `TikZiT`, un éditeur graphique pour `TikZ` très simple. Si pour les dessins techniques, notamment géométriques, il me semble préférable de produire directement un code `TikZ`, pour d'autre cas, plus graphiques, un logiciel comme celui-là peut se révéler utile. La possibilité

d'utiliser un logiciel avec interface graphique, **WYSIWYG**, est aussi un bon outil pour aider les moins à l'aise avec l'édition de code informatique.

TikZiT est un logiciel libre et multiplateformes. Même si je n'ai pas utilisé intensivement ce logiciel, sa simplicité d'interface et la clarté de sa documentation me semblent être des gages de qualité.

Bref, la multiplicité des outils autour de nos logiciels favoris est une grande richesse (même si elle est tellement foisonnante qu'il est parfois difficile de s'y retrouver), et ce logiciel fait partie de ceux que je pourrai conseiller à l'avenir.

Maxime Chupin

## Étienne Ghys à la séance solennelle de rentrée des cinq académies

Le mardi 25 octobre 2022, lors de la séance solennelle de rentrée des cinq académies, Étienne Ghys a donné un discours intitulé « La géométrie des lettres de l'alphabet ». Étienne Ghys est un mathématicien français, secrétaire perpétuel (première division) de l'Académie des sciences. Il est directeur de recherche émérite au **CNRS**, et est principalement connu pour sa recherche en géométrie et sur les systèmes dynamiques, ainsi que pour ses travaux de vulgarisation.

Lors de cette séance, il ne déroge pas à son habitude d'exposer avec une grande clarté, mais cette fois-ci, c'est la typographie qui est à l'honneur. Nous vous invitons à visionner son discours ici :

[https:](https://www.youtube.com/live/rpWyejz1FRU?feature=share&t=3288)

[//www.youtube.com/live/rpWyejz1FRU?feature=share&t=3288](https://www.youtube.com/live/rpWyejz1FRU?feature=share&t=3288)

Il y parle d'art, d'histoire, de technique, de mathématiques et de figure géométriques. C'est un exposé passionnant, et évidemment il rend hommage à Donald E. Knuth et notamment de ses travaux sur la typographie numérique (*Digital Typography*). Il y parle ainsi de METAFONT et de courbes de Bézier. Il y fait même la promotion, certes à lire entre les lignes, des logiciels libres.

Bref, prenez vingt minutes de votre temps, et nous espérons que vous passerez du bon temps.

Maxime Chupin

## Un appel à candidatures

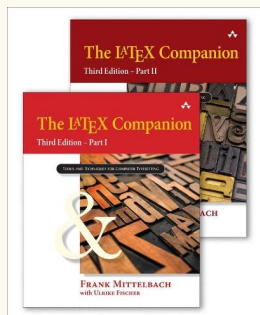
La Cité internationale de la langue française, sise au château de Villers-Cotterêts, en partenariat avec le centre national des Arts plastiques et avec l'Imprimerie nationale, lance un appel à candidatures pour une résidence artistique dédiée à la création typographique. Cette résidence se déroulera à partir d'octobre prochain.

<https://www.cnap.fr/annonces/appel-candidatures-pour-une-residence-artistique-dediee-la-creation-typographique>

📅 Date limite de dépôt du dossier : 3 juillet 2023

## $\LaTeX$ Companion, 3<sup>e</sup> édition, 2023

Ça y est, le  $\LaTeX$  Companion, 3<sup>e</sup> édition, de Frank Mittlebach et Ulrike Fischer, aux éditions Addison-Wesley Professional est disponible à la commande !



Le  $\LaTeX$  Companion est une référence très importante pour  $\LaTeX$ . Sa première édition écrite par Michel Goossens, Frank Mittlebach et Alexander Samarin date de 2000 et était un ouvrage massif de 559 pages. La deuxième édition de Michel Goossens, Frank Mittlebach, Johannes Braams, David Carlisle et Chris Rowley parue en 2004 fut encore plus massive avec 1120 pages.

Ces deux premières éditions ont été des références faisant un état des lieux de  $\LaTeX$ , décortiquant en profondeur ses mécanismes et ses packages les plus importants.

C'est désormais une troisième édition, de nouveau encore plus massive puisqu'elle est en deux parties de respectivement 976 et 1008 pages. Depuis 2004 le monde de  $\LaTeX$  a considérablement changé : des packages fondamentaux sont apparus (`biblatex`, `fontspec`, `hyperref`, `mathtools`, `siunitx`, `tikz`, `tcolorbox`, etc.) ; les nouveautés apportées par le développement de  $\LaTeX$ 3 ; les nouveautés des moteurs  $X_{\LaTeX}$  et  $\text{Lua}\TeX$  ; la gestion des fontes TrueType et OpenType ; etc.

Nous reproduisons ici un sommaire des chapitres :

### Part I

- Chapter 1: Introduction
- Chapter 2: The Structure of a  $\LaTeX$  Document
- Chapter 3: Basic Formatting Tools, Paragraph Level
- Chapter 4: Basic Formatting Tools, Larger Structures
- Chapter 5: The Layout of the Page
- Chapter 6: Tabular Material
- Chapter 7: Mastering Floats
- Chapter 8: Graphics Generation and Manipulation
- Chapter 9: Font Selection and Encodings

### Part II

- Chapter 10: Text and Symbol Fonts

- Chapter 11: Higher Mathematics
- Chapter 12: Fonts in Formulas
- Chapter 13: Localizing Documents
- Chapter 14: Index Generation
- Chapter 15: Bibliography Generation
- Chapter 16: Managing Citations
- Chapter 17:  $\LaTeX$  Package Documentation Tools

### Appendix

- Appendix A:  $\LaTeX$  Overview for Preamble, Package, and Class Writers
- Appendix B: Tracing and Resolving Problems
- Appendix C: Going Beyond
- Bibliography

Rappelons que la deuxième édition a été traduite en français sous la supervision de Jacques André (de l'association GUTenberg, groupe francophone des utilisateurs de  $\TeX$ ), Benoît Bellet, Jean-Côme Charpentier, Jean-Michel Hufflen et Yves Soulet.

Un projet d'envergure pour l'association pourrait être d'organiser collectivement la traduction de ce document. Qui est partant ?

Maxime Chupin

## (all)T<sub>E</sub>X au café

Divers utilisateurs de T<sub>E</sub>X, de L<sup>A</sup>T<sub>E</sub>X et autres logiciels apparentés se sont retrouvés, le 12 mai dernier, au café du Père tranquille, dans le centre de Paris. Dans une ambiance conviviale, de nombreux sujets furent abordés, notamment à l'occasion de différents tours de table qui furent nécessaires à mesure de l'arrivée de tel ou tel, de manière à ce que les participants se présentent. Fait notable, loin d'être redondantes, ces présentations réitérées firent apparaître différents aspects de nos activités autour de ces logiciels. Ainsi, l'un des participants témoigna de sa découverte de T<sub>E</sub>X, dans le milieu des années 80, sur un ordinateur Atari ST<sup>53</sup>... ce qui en laissa certains pantois.

Parmi les sujets qui furent abordés, ou simplement effleurés, et dans un inventaire à la Prévert (mais composé avec L<sup>A</sup>T<sub>E</sub>X), on notera :

- l'utilité de la coloration ligne à ligne, qui offre une véritable aide à la lecture pour des élèves en difficulté mais pose d'importants problèmes à ceux qui composent ces textes<sup>54, 55</sup> ;
- la nécessaire transcription des cours de Jean Douchet<sup>56</sup>, vaste entreprise dans laquelle Alain Crémieux s'est lancé ;
- l'utilisation de L<sup>A</sup>T<sub>E</sub>X pour relater un voyage à vélo (l'influence de la fréquence de pédalage sur le temps de compilation n'a malheureusement pas été évoquée) ;
- κ,,ἰαω — la composition du syriaque estrangelo (vraisemblablement sans relation avec le *travelogue* précédent<sup>57</sup>) ;
- la poésie concrète et l'utilité de T<sub>E</sub>X pour la mettre en page ;
- le volume de la documentation d'un bon logiciel de *product lifecycle management*, si important qu'il la rend parfois inutilisable<sup>58</sup> ;
- l'atmosphère des planètes, qui donne lieu à un vaste projet éditorial auquel participe l'un des membres de l'association ;
- la grapholinguistique, utile entreprise à laquelle se consacre Yannis Haralambous ;
- le regret de ne pas voir MusiX<sub>T</sub>E<sub>X</sub> communiquer en midi,
- la pertinence des propos de Till Tantau sur le graphisme<sup>59</sup> ;

53. Que ceux qui ont utilisé T<sub>E</sub>X ou L<sup>A</sup>T<sub>E</sub>X sur des systèmes d'exploitation autres qu'Unix (ou assimilés), MacOS ou Windows se manifestent ! Un exemple : la documentation de T<sub>E</sub>X sur RiscOS manque cruellement et il incombe à l'association d'y suppléer.

54. Nous avons retrouvé ce fil qui traite de ce problème : <https://texnique.fr/osqa/questions/4987/textes-pour-dyslexiques>.

55. Sur ce thème, rappelons l'existence de la fonte Luciole, spécialement conçue pour les malvoyants : <https://www.luciole-vision.com>.

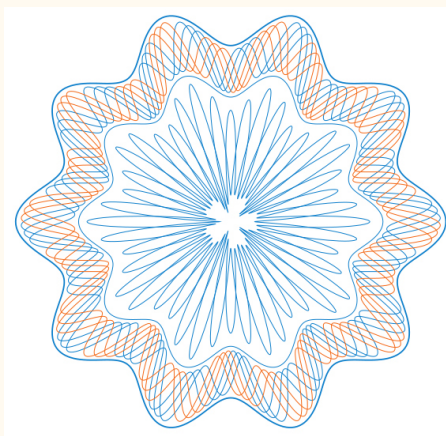
56. Jean Douchet (1929 – 2019) est un historien et critique du cinéma. Il a enseigné à l'IDHEC et à la FÉMIS. Il analysait les films en public, à l'issue de la projection de ceux-ci, notamment à la Cinémathèque française où il anima un ciné-club très réputé. Son éloquence généreuse et sa connaissance approfondie du cinéma drainait un public nombreux. Il laisse une œuvre écrite assez brève, inversement proportionnelle à son influence sur des générations de cinéastes et de cinéphiles. La publication de ses cours contribuera à la préservation et à la diffusion de son héritage intellectuel. Nous l'appelons de nos vœux.

57. Nous recevons à l'instant une communication de la Fédération internationale de cyclotourisme confirmant nos craintes : ce loisir n'est plus guère pratiqué dans les contrées où le syriaque est parlé — *o tempora, o mores*...

58. Ce fut l'occasion de changer de point de vue sur la documentation de TikZ : en comparaison, celle-ci est d'une longueur tout à fait raisonnable.

59. Nous vous renvoyons aux très utiles *Guidelines on Graphics* qui figurent en section I.7 de la documentation de TikZ (package `pgf`).

FIGURE 47 – Un guillochis



- le problème posé par la pénurie de symboles, à laquelle on peut être confronté quand un projet éditorial embrasse un très vaste ensemble de connaissances, et les moyens d’y remédier ;
- les cartes thermiques, ou *heatmaps*, chères aux actuaires et réalisables avec R et, par exemple, TikZ (METAPOST ou PSTricks peuvent également être utilisés) ;
- l’utilisation du package `glossaries` et la pérennité de celui-ci ;
- et la parution en anglais de la troisième édition du *L<sup>A</sup>T<sub>E</sub>X Companion*, qui donna lieu à une conversation sur une éventuelle traduction française (voir page 76).

Enfin, de très beaux guillochis, réalisés à l’aide de lua et de TikZ pour un package à venir, furent présentés à l’assistance pour son plus grand plaisir ; voir la figure de la présente page. Les conversations furent animées et ce beau moment illustra, s’il en était besoin, la vitalité de la communauté francophone.

Patrick Bideault

## ***Gustave*, un irrégulomadaire de poésie**

La rédaction de la présente *Lettre* ne peut qu’apprécier le travail de son homologue qui préside aux destinées de *Gustave*. Cet irrégulomadaire gratuit propose, à chaque livraison, des poèmes contemporains de grande qualité. Il est disponible sur simple abonnement.

<https://www.gustavemagazine.com>

## **Deux expositions**

Le samedi 17 juin à 16 heures,

quelques membres de l’association ont prévu d’aller visiter l’exposition Gutenberg à la bibliothèque nationale. N’hésitez pas à vous joindre à cette visite : il vous suffit d’écrire au secrétariat pour convenir d’un rendez-vous.

<https://www.bnf.fr/fr/agenda/imprimer-leurope-de-gutenberg>

Par ailleurs, l’atelier-musée de l’imprimerie, sis à Malesherbes, dans le nord du Loiret, présente jusqu’au 24 septembre prochain une exposition consacrée

au graphiste français Marcel Jacno (1904-1989) dont l'œuvre est restée dans la mémoire collective.

<https://a-mi.fr/exposition-marcel-jacno-typo-graphiste/>

## À la radio

DU 17 au 20 avril dernier, chacun des épisodes de l'émission *Le Cours de l'histoire*, diffusée sur France Culture, était consacré à l'écriture. On y traite de l'élamite linéaire (un système d'écriture récemment déchiffré), de Gutenberg (à l'occasion de l'exposition que lui consacre la bibliothèque nationale, voir page ci-contre), de Garamont<sup>60</sup> et du braille.

Elles sont à retrouver ici :

<https://www.radiofrance.fr/franceculture/podcasts/serie-histoires-d-ecritures>

Par ailleurs, la même radio a proposé, les 18 et 25 avril derniers, deux nuits consacrées au livre, aux librairies et à leur histoire.

Ce très précieux ensemble d'émissions radiophoniques est à retrouver ici :

<https://www.radiofrance.fr/franceculture/podcasts/selectio-n-librairies-du-manuscrit-au-livre-de-poche>

<https://www.radiofrance.fr/franceculture/podcasts/serie-nuit-de-la-librairie-2-2>



## « INTRODUCTION À LUAL<sub>A</sub>T<sub>E</sub>X »



L'association GUTenberg organise un exposé

« Introduction à Lua<sub>A</sub>T<sub>E</sub>X »

par Denis Bitouzé

le **jeudi 22 juin de 20 heures à 21 heures**  
en visio-conférence<sup>a</sup>.

<sup>a</sup>. Les informations de connexion seront fournies sur les canaux habituels peu avant la conférence.



60. Signalons à nos lecteurs la parution, l'an dernier aux éditions des cendres, de *Claude Garamont, typographe de l'humanisme*, par Rémi Jimenes.

## 🌀 RÉBUS

### 🔑 Solution du rébus de la *Lettre* précédente

L'homme en grandissant change de caractère.

### 🔑 Rébus (*un toast*)

Mademoselle, à voire !

*Solution dans la prochaine Lettre*



## 🌀 ACRONYMES

**CA** Conseil d'Administration

**CMS** *Content Management System* (système de gestion de contenu)

**CNRS** Centre National de la Recherche Scientifique

**CTAN** *Comprehensive T<sub>E</sub>X Archive Network* (réseau complet d'archives T<sub>E</sub>X)

**DOI** *Digital Object Identifier* (identifiant numérique d'objet)

**EBNF** *Extended Backus-Naur Form* (forme de Backus-Naur étendue)

**PDF** *Portable Document Format* (format de document portable)

**RGB** *Red Green Blue* (rouge vert bleu)

**TikZ** *TikZ ist kein Zeichenprogramm* (TikZ n'est pas un programme de dessin)

**TUG** *T<sub>E</sub>X User Group* (groupe international d'utilisateurs de T<sub>E</sub>X)

**WYSIWYG** *What You See Is What You Get* (ce que vous voyez est ce que vous obtenez)

**YAML** *YAML Ain't Markup Language* (YAML n'est pas un langage de balisage (acronyme récursif))

La rédaction tient à remercier MFerrer pour les erreurs qu'il a remarquées dans la *Lettre* 49; ces erreurs ont été corrigées. Si vous en remarquez d'autres, merci de nous les signaler par courriel à [secretariat@gutenberg-asso.fr](mailto:secretariat@gutenberg-asso.fr).

Le contenu de cette *Lettre* a été figé le 6 juin 2023. Elle a ensuite été corrigée, puis publiée le 22 juin 2023.

Par leurs textes, leurs suggestions ou leurs corrections, et par ordre alphabétique de patronymes, ont contribué à cette *Lettre* : Patrick Bideault, Denis Bitouzé, Céline Chevalier, Maxime Chupin, François Druel, Yvon Henel, Jean-Michel Hufflen, Yannick Tanguy & Flora Vern.



# GUTenberg

Association GUTenberg  
15 rue des Halles – BP 74  
75001 Paris  
France  
secretariat@gutenberg-asso.fr

**Site Internet :** <https://www.gutenberg-asso.fr/>

**Cahiers :** <https://cahiers.gutenberg-asso.fr/>

**Lettre :** <https://lettre.gutenberg-asso.fr/>

**Problèmes techniques :**

**liste d'entraide :** <https://www.gutenberg-asso.fr/-Listes-de-diffusion->

**site de questions et réponses :** <https://technique.fr/>

**foire aux questions :** <https://faq.gutenberg-asso.fr/>

Cette association est la vôtre : faites-nous part de vos idées, de vos envies, de vos préoccupations à l'adresse [secretariat@gutenberg-asso.fr](mailto:secretariat@gutenberg-asso.fr).  
Adhérents, vous pouvez aussi échanger sur la vie de l'association sur la liste de diffusion [adherents@gutenberg-asso.fr](mailto:adherents@gutenberg-asso.fr).

## ADHÉSION À L'ASSOCIATION

- Les adhésions sont à renouveler en début d'année pour l'année civile.
- Les administrations peuvent joindre un bon de commande revêtu de la signature de la personne responsable; les étudiants doivent joindre un justificatif.

### Tarifs <sup>61</sup> 2023

Les membres de GUTenberg peuvent adhérer à l'association internationale, le TUG, et recevoir son bulletin TUGboat à un tarif préférentiel <sup>62</sup> :

Type d'adhésion	Prix
Personne physique	30 €
Personne physique + adhésion TUG	95 € (= 30 € + 65 €)
Personne physique à tarif réduit	15 €
Personne physique à tarif réduit + adhésion TUG	55 € (= 15 € + 40 €)
Association d'étudiants	65 €
Organisme ou association à but non lucratif	130 €
Personne morale à but lucratif	229 €

## Règlements

Les règlements peuvent s'effectuer par :

- **virement bancaire** <sup>63</sup> (IBAN : FR76 1870 7000 3003 0191 3568 475)

**Veillez à bien indiquer vos nom et prénom dans les références du virement !**

- Paypal<sup>63</sup> : <https://www.gutenberg-asso.fr/?Adherer-en-ligne>
- bulletin et chèque<sup>63</sup> : <https://www.gutenberg-asso.fr/?Adherer-a-l-association>

**La Lettre GUTenberg**  
Bulletin irrégulomestriel de l'association GUTenberg

**Directeur de la publication :** Patrick Bideault  
**Comité de rédaction :** Patrick Bideault, Denis Bitouzé, Céline Chevalier, Maxime Chupin & Yvon Henel

**Adresse de la rédaction :** Association GUTenberg  
15 rue des Halles – BP 74  
75001 Paris

ISSN : 2742-6149 (version numérique)

61. Dans le tableau suivant, une personne physique à tarif réduit est étudiant, demandeur d'emploi ou plus largement toute personne non redevable de l'impôt sur le revenu (sur présentation d'un justificatif); un organisme peut être doté ou non de la personnalité morale : laboratoire de recherche public, etc.

62. En tarif normal, 65 € au lieu de 85 \$; en tarif étudiant, 40 € (au lieu de 55 \$).

63. Nous vous remercions de **privilégier** le **virement bancaire**.