

La

**L
E
T
T
R
E**

Numéro 41 – Décembre 2020

INDEX — SOMMAIRE

Journée GUTenberg 2020	2
Assemblée générale de GUTenberg	2
Conseil d'administration de GUTenberg	5
Décomposition en série de FOURIER d'un caractère	6
Campus du libre, édition 2020	16
Compte rendu de lecture	17
Association GUTenberg (tout sur l')	18

Éditorial

Nouvellement élus, c'est avec émotion que nous écrivons les premières lignes de ce billet : si la responsabilité que vous avez placée en nos mains nous honore, nous mesurons l'ampleur de la tâche qui désormais nous incombe.

Nous saluons ici notre prédécesseur, Jérémy JUST, sans lequel nous ne nous serions pas engagés pour l'association. Il l'a dirigée dans une période difficile, il a été réélu et siège désormais au CA, où il apporte une aide précieuse. Qu'il soit ici remercié.

Vous lirez dans ces pages le compte rendu de l'assemblée générale : le 14 novembre dernier, vous participâtes nombreux à ces rencontres en ligne. Nous y voyons un signe de vigueur de notre association.

Une semaine plus tard, les nouveaux administrateurs se sont également réunis en ligne ; ce fut une prise de contact aussi plaisante que constructive. Elle vous est rapportée ci-après, avant un article qui réjouira les amateurs de METAPOST ; Maxime CHUPIN, membre du CA et principal initiateur de la présente Lettre, en est l'auteur.

Nous espérons que vous apprécierez cette quarante-et-unième livraison de la Lettre GUTenberg, que par simplicité nous vous adressons par courriel.

Nous tenons ici à rendre hommage à deux adhérents qui ont récemment rendu de grands services à l'association. En premier, Arthur REUTENAUER, ancien président et actuellement vice-président du TUG, a bien voulu présider l'assemblée générale et être le garant de l'élection du conseil d'administration. Il s'est acquitté de ces tâches avec maestria et nous lui en savons gré d'autant que, quelques jours plus tard, il organisait l'élection du bureau ! Par ailleurs, Arthur s'est montré intéressé par notre projet de refonte des statuts et a pris contact en ce sens avec Flora VERN, notre trésorière, qui est juriste et dont l'expertise nous est précieuse. Il va sans dire que les adhérents seront consultés au sujet de cette réforme des statuts.

GUTenberg

En second, Yannick TANGUY dont vous avez lu dans la précédente Lettre l'article relatif à la base de données des adhésions, a été élu au conseil d'administration; ex-æquo avec deux autres candidats, il leur a immédiatement cédé sa place, dans un geste d'une rare élégance. Conformément à ce qu'il avait expliqué au sujet de sa candidature, il a tout de suite proposé son aide au CA pour la migration de cette base sur le nouveau serveur que nous allons mettre en place, et a fait part de son intérêt pour le logiciel de gestion d'association que nous allons déployer. Désintéressée et toujours constructive, l'attitude exemplaire de Yannick nous est d'un grand soutien et nous l'en remercions vivement.

Vous le lirez plus loin : le nouveau conseil d'administration est au travail, et nous en sommes très heureux. La discussion sur les projets 2020–2021 n'ayant pas pu avoir lieu à l'assemblée générale du 14 novembre faute de temps, nous vous convierons à un temps d'échange en ligne à ce sujet, probablement courant janvier. Et c'est avec un grand plaisir qu'au printemps prochain, pour un premier bilan d'étape, nous retrouverons les adhérents pour l'assemblée générale d'une association bien vivante, utile et déterminée.

Associativement,

Patrick BIDEAULT, président 2020-2022,
et Céline CHEVALIER, vice-présidente 2020-2022.

JOURNÉE GUTENBERG 2020

Le 14 novembre a eu lieu l'assemblée générale de l'association (en visioconférence). Elle a été doublée d'une journée Gutenberg grâce aux interventions de trois orateurs invités par JérémY JUST. Le matin et le début d'après-midi ont ainsi été consacrés à trois exposés sur le thème « Du caractère à la reliure : le numérique fait aussi de beaux livres », qui ont été très bien accueillis :

- « De la pâte à modeler vectorielle, ou comment naissent les polices de caractère » par Émilie RIGAUD;
- « Mettre en page un roman avec la famille T_EX » par Thomas SAVARY;
- « Papiers marbrés en PStricks » par Aubrey JAFFER.

Nous reviendrons en détail sur ces interventions dans la prochaine Lettre, prévue pour le mois de janvier, et vous pouvez trouver le compte rendu de l'assemblée générale ci-après.

Céline CHEVALIER

COMPTE RENDU DE L'ASSEMBLÉE GÉNÉRALE DU 14 NOVEMBRE 2020

Introduction

L'assemblée générale s'est déroulée cette année dans des circonstances particulières, en raison non seulement de l'impossibilité de se réunir physiquement mais aussi de la relative somnolence qui avait caractérisé l'association pendant plusieurs années. Le conseil d'administration, en particulier, devait être renouvelé dans sa totalité, les mandats des différents élus étant arrivés à échéance en 2015 ou 2017, suivant les personnes.

Du fait de cette situation exceptionnelle, les débats ne se sont pas toujours déroulés dans la sérénité qu'on était en droit d'espérer, et pour cette raison je n'écrirai pas tout ce qui a été dit. J'invite les adhérents qui auraient des questions à me contacter personnellement (arthur.reutenauer@normalesup.org).

Les discussions ont eu lieu sur la plate-forme BigBlueButton, de vive voix et sur le *chat*, et les votes annoncés à l'ordre du jour sur l'outil Bélénios de vote en ligne. Les scrutins

furent ouverts jeudi 12 novembre au soir pour que le plus grand nombre d'adhérents puissent voter, et fermés au fur et à mesure des débats.

Ordre du jour

La séance est ouverte à 15 heures avec 35 personnes en ligne sur les 104 adhérents à jour de leur cotisation. Les statuts de l'association n'imposent pas de quorum. L'ordre du jour est :

- Scrutin au sujet de la dissolution de l'association;
- Présentation des bilans 2014–2019 et vote;
- Élection des membres du conseil d'administration;
- Discussion et votes sur les projets 2020–2021;
 - Refonte des documents de communication et du site Web;
 - *Cahiers GUTenberg*;
 - Organisation d'ateliers en visioconférence (FAQkathon & formations);
 - Attentes de la communauté (discussion ouverte).

Scrutin sur la dissolution

La proposition soumise au vote est :

Souhaitez-vous la dissolution de l'association GUTenberg?

Elle est assortie du paragraphe suivant :

Notez que si le résultat est favorable à la dissolution, il faudra décider de l'utilisation des fonds associatifs conformément à l'article 9 de la loi du 1^{er} juillet 1901. En revanche, si le résultat est en faveur du maintien de l'association, celui-ci reste conditionné à l'élection de suffisamment de volontaires au CA.

Le scrutin est clos à 15 heures 20 après une courte discussion. Le nombre de votants est 54; 5 votes pour et 49 contre. Il n'y a aucun bulletin blanc.

Bilans

Le bilan moral et le rapport financier sont présentés par JérémY Just. Ils couvrent les années 2014 à 2019 incluses. Ils ont été publiés dans la *Lettre GUTenberg* n° 40, et la présentation correspondante est disponible sur le site web : <https://www.gutenberg.eu.org/GUT2020>

Au cours de cette période, l'association a coorganisé deux jours de conférences à l'ENSSIB en 2017. Un sondage sur les manifestations futures de l'association est en cours. Il est rappelé qu'il existe deux publications, la *Lettre GUTenberg* et les *Cahiers GUTenberg*.

Les DVD de la T_EX Collection, fabriqués par l'association DANTE en Allemagne, sont commandés chaque année mais ne sont expédiés aux adhérents qu'à la faveur d'autres envois (relances de cotisation ou *Lettre GUT*), ce qui entraîne une certaine irrégularité. Ceux qui souhaitent les recevoir rapidement peuvent en faire la demande. Le reliquat est distribué lors de manifestations externes.

L'association est un point de contact pour des initiatives, comme les motifs de césure et les demandes de formation. De nombreuses traductions de documentation ont été produites, mais elles sont peu mises en valeur. La présentation se conclut par un appel à bonnes volontés pour le développement de l'infrastructure informatique.

Le bilan comptable du début 2014 à la fin 2019 fait apparaître un résultat excédentaire de 1889 €, pour un actif de 28948 €. Le bilan détaillé et le compte de résultat ont été publiés dans la *Lettre GUTenberg* n° 40.

Il est alors procédé au vote sur ces bilans. Il y a à nouveau 54 votants, le bilan moral reçoit 36 voix pour, 13 contre et 5 bulletins blancs ; le rapport financier 35 pour, 14 contre et 5 blancs.

Élection du conseil d'administration

L'assemblée générale se prépare alors à élire le nouveau conseil d'administration. Tous les douze postes sont à pourvoir, il y a quatorze candidats, et on entame la discussion par un vote sur le mode d'élection. La question est de savoir si le dernier alinéa de l'article 5, qui dispose que « les décisions de l'Assemblée générale (ordinaire ou extraordinaire) sont prises à la majorité absolue des membres présents ou représentés » s'applique à l'élection des membres du CA. Si c'est le cas, cela reviendrait à déclarer élus les candidats qui auront reçus une majorité absolue du nombre de votants, dans la limite de douze.

Cette proposition est soumise au vote grâce à l'outil de sondage BigBlueButton. Elle reçoit 22 voix pour et 9 contre.

Les 14 candidats peuvent alors se présenter, chacun à leur tour. À l'issue de cette présentation, on laisse dix minutes pour que tout le monde puisse finir de voter, la plate-forme de vote en ligne permettant de changer un vote antérieur tant que le scrutin n'est pas clos.

Le scrutin est finalement clos à 18 heures 55, et le dépouillement, automatique grâce à l'application utilisée, Belenios, est presque instantané. Il y a 56 votants, la majorité absolue est donc de 29 voix.

Résultat par ordre décroissant du nombre de voix :

Céline Chevalier	48
Flora Vern	48
Patrick Bideault	44
Yvon Henel	42
Jean-Michel Hufflen	42
Denis Bitouzé	40
Maxime Chupin	40
Éric Guichard	33
Jérémy Just	33
Robert Alessi	30
Christian Hinqué	29
Yannick Tanguy	29
Yoan Tournade	29
Jacques Abada	14
Bulletin blanc	1

Le tableau ci-dessus met en avant un défaut du mode de scrutin, puisque sur les treize candidats recevant une majorité absolue, les trois derniers ne peuvent pas être départagés. L'un d'entre eux, Yannick Tanguy, propose alors spontanément et généreusement de se désister. Qu'il en soit ici remercié par écrit.

L'assemblée tombe d'accord pour repousser à plus tard la discussion sur les projets 2020–2021 et laisse l'initiative au CA. La séance est levée à 19 heures 05.

Le président de séance,

Arthur REUTENAUER

COMPTE RENDU DU CONSEIL D'ADMINISTRATION DU 21 NOVEMBRE 2020

Présents : Robert ALESSI, Patrick BIDEAULT, Denis BITOUZÉ, Céline CHEVALIER, Éric GUICHARD, Yvon HENEL, Christian HINQUE, Jean-Michel HUFFLEN, Jérémy JUST, Yoan TOURNADE, Flora VERN.

Excusé : Maxime CHUPIN

Suite à leur élection lors de l'assemblée générale du 14 novembre 2020, les membres du conseil d'administration se sont réunis par visioconférence le samedi 21 novembre 2020.

Ce fut l'occasion d'un premier contact en vue de faire connaissance, de définir les projets prioritaires et de les lancer.

Ont tout d'abord été élus au bureau les membres suivants :

Président : Patrick BIDEAULT

Vice-présidente : Céline CHEVALIER

Secrétaire : Denis BITOUZÉ

Secrétaire adjoint : Maxime CHUPIN

Trésorière : Flora VERN

Trésorier adjoint : Yvon HENEL

Patrick BIDEAULT entame ensuite la discussion en abordant le sujet de la gestion quotidienne de l'association et en proposant l'outil Dolibarr, qu'il a déjà testé. La base de données élaborée par Yannick TANGUY resterait bien sûr en utilisation le temps nécessaire à la migration, ce dernier étant d'ailleurs d'accord pour y participer.

La migration du serveur est également en projet. L'objectif principal est le regroupement de tous les services sur le même serveur, dans un souci de simplification. La maintenance doit être rendue plus aisée et les savoirs et savoir-faire faciles à transmettre entre CA successifs. L'étape suivante sera la restructuration du site web et la mise à disposition de services supplémentaires. Yoan TOURNADE se propose comme coordinateur de ces projets. La solution d'hébergement adoptée et le changement éventuel du nom de domaine (en gardant une redirection depuis le nom actuel) sont en discussion.

Un autre projet important discuté lors de cette réunion est la réforme des statuts, chantier coordonné par Flora VERN et sur lequel les adhérents seront consultés début 2021.

Jean-Michel HUFFLEN étudie actuellement une reprise des « Cahiers GUTenberg » avec une thématique élargie et un comité de lecture renouvelé. T_EX, L^AT_EX, ConT_EXt, LuaT_EX vont bien sûr rester à l'honneur, ainsi que la typographie en général, sans oublier la typographie des mathématiques et celle de la musique. D'autres articles plus techniques pourront traiter des avancées bibliographiques, d'autres plus ouverts sur les sciences sociales pourront aborder les évolutions des métiers du livre ou étudier les problématiques suscitées par le livre électronique. Il s'agit d'un vaste chantier, mais dont l'avancement nous fait présager la sortie du prochain numéro au printemps 2021, et d'un numéro plus thématique à l'automne. Nous espérons pouvoir maintenir cette vitesse de croisière par la suite. Il s'agit aussi d'un chantier ouvert : si vous avez des propositions d'articles dans vos cartons, vous pouvez les envoyer à cahiers@gutenberg.eu.org.

Au niveau administratif, l'association est actuellement hébergée au sein d'une unité de recherche dirigée par Thierry BOUCHE, mais dont il quitte la direction cette année. Elle

va donc devoir changer de siège social et la recherche d'une nouvelle domiciliation est lancée. Le changement de domiciliation bancaire est également à l'étude.

Tous ces projets seront mis en œuvre dans le cadre de groupes de travail, utilisant abondamment le serveur Git de l'association et des listes de discussion (archivées) mises en place sur le serveur. D'autres moyens de communication, travail et gestion de projet sont encore en discussion. L'un des objectifs est de faciliter la passation d'un CA à l'autre.

Nous prévoyons également de communiquer régulièrement avec les adhérents et de leur permettre de débattre des questions associatives sur une liste de discussion dédiée à créer, la liste `gut@ens.fr` n'étant pas le lieu pour cela, comme cela a plusieurs fois été demandé par des abonnés par le passé. Cette dernière liste a d'ailleurs vocation à être migrée à terme sur le nouveau serveur de l'association.

Enfin, les changements apportés au conseil d'administration et au bureau ont été déclarés (en ligne) à la préfecture, afin de retrouver un fonctionnement conforme à la législation. À l'heure de publier ces lignes, nous attendons la réponse de la préfecture.

Céline CHEVALIER

DÉCOMPOSITION EN SÉRIE DE FOURIER D'UN CARACTÈRE AVEC L^AT_EX ET MPLIB

Le vidéaste 3Blue1Brown¹, vulgarisateur mathématique sur YouTube, a réalisé une vidéo où il illustre la décomposition de Fourier d'un chemin fermé par des animations de mécanismes d'engrenages de cercles mis bout à bout. Le résultat est magnifique et envoûtant (voir figure 1).

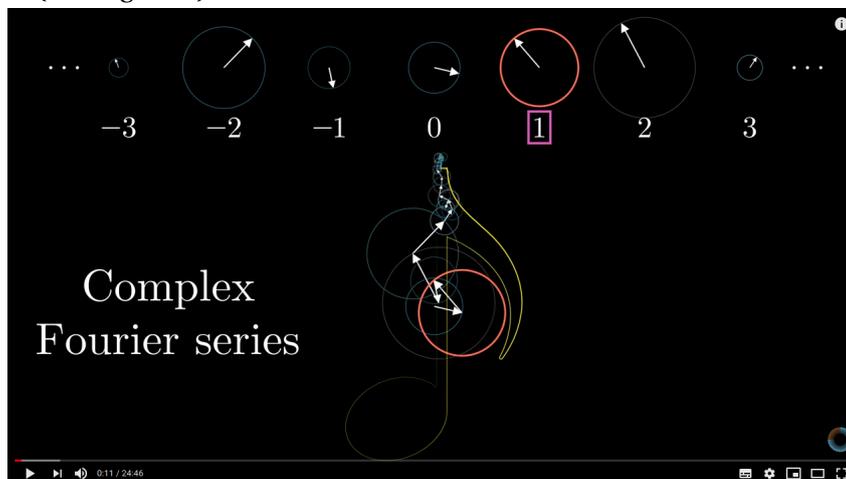


FIGURE 1 – Vidéo *Mais qu'est-ce qu'une série de Fourier? Du transfert thermique à des dessins avec des cercles* du vidéaste 3Blue1Brown sur YouTube.

Il n'est pas facile de décrire avec des mots ces animations, mais il s'agit de mettre bout à bout des cercles de différents diamètres avec des vecteurs inscrits qui tournent à différentes vitesses, et l'extrémité de cette ligne brisée trace la courbe fermée considérée (la note de musique sur la figure 1).

Ce visionnage m'a donné l'envie² de réaliser cela avec nos outils préférés et plus particulièrement avec L^AT_EX et METAPOST, lui-même inclus dans L^AT_EX *via* la librairie Mplib. Je me suis aussi dit que cela serait intéressant de réaliser ces animations avec le contour d'un glyphe d'un caractère (d'une seule partie évidemment, connexe). C'est donc à ce chantier que je me suis attelé.

1. <https://www.youtube.com/watch?v=-qgreAUppwM>

2. Pour un cours du traitement du signal que je dispense en troisième année d'étude supérieures.



Avertissement : Cet article parle de \LaTeX , de Lua, et de METAPOST [5]. Pour bien le comprendre, il faudra avoir quelques notions de ces trois aspects. Pour une bonne introduction à Lua \LaTeX je conseille la lecture des cahiers GUTenberg dédiés [2]. Pour une bonne introduction à la programmation en Lua, allez voir *Programming in Lua* [7] (lua.org). Je ne peux malheureusement pas revenir sur toutes les commandes ou fonctions des divers langages dans cet article; cependant, j'ai essayé de dégager l'essentiel.

Le principe mathématique de l'affaire

On considère donc une courbe fermée dans \mathbf{R}^2 que l'on peut considérer comme une fonction $f : \mathbf{R} \rightarrow \mathbf{C}$ périodique. On considère que la période est égale à 1. Sans rentrer dans les détails, la décomposition en série de Fourier de f est :

$$\forall t \in [0, 1], \quad f(t) = \sum_{n=-\infty}^{+\infty} c_n(f) e^{in2\pi t},$$

où

$$c_n(f) = \int_{-1/2}^{1/2} f(t) e^{-in2\pi t} dt.$$

Numériquement, on travaillera avec des versions discrètes de cette décomposition en série de Fourier. Considérons deux entiers N et M assez grands et M pair. Dans le monde discret, nous n'aurons plus la fonction f continue mais des échantillons le long du chemin, que l'on notera (f_1, f_2, \dots, f_N) où les $f_i \in \mathbf{C}$. On a alors :

$$\forall t \in [0, 1], \quad f(t) \simeq \sum_{n=-M/2}^{M/2} \tilde{c}_n(f) e^{in2\pi t}, \quad (1)$$

où

$$\tilde{c}_n(f) = \sum_{k=0}^N \frac{1}{N} f_{k+1} e^{-i\frac{2nk}{N}}. \quad (2)$$

On appellera les $\tilde{c}_n(f)$ les coefficients de Fourier qui sont au nombre de $M + 1$.

Géométriquement, on peut voir la relation (1) comme une somme de vecteurs de \mathbf{R}^2 (donc mis bout à bout) de norme le module du nombre complexe $\tilde{c}_n(f) e^{in2\pi t}$ et, comme orientation, son argument. Ainsi, quand t parcourt l'intervalle $[0, 1]$, ces vecteurs bout à bout tournent et l'extrémité du dernier vecteur dessine la courbe fermée³ que l'on a décomposée.

Récupération du contour en un ensemble de points de \mathbf{R}^2

Il nous faut donc construire, à partir d'un contour d'un glyphe, la suite (f_1, f_2, \dots, f_N) présentée ci-dessus.

Grâce à METAPOST

Dans un premier temps, il nous faut obtenir une discrétisation du contour fermé d'un glyphe donné. METAPOST, avec le format MetaFun [3], permet de faire cela assez facilement. Pour l'exemple, nous allons prendre le glyphe f , un nombre de points de 500 pour la discrétisation, et un certain facteur homothétique fixé à 0, 1 pour l'affichage dans ce document :

3. Ou plutôt une approximation du contour du glyphe.

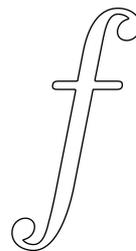
```

fontmapfile "=lm-ec.map";
picture lettre;
path contourLettre;
lettre := glyph "f" of "ec-lmri10"; % les lettres sont en général
                                   % composées de plusieurs parties

path p;
nbrPoints := 500;
scale := 0.1;
beginfig(1);
  for item within lettre: % on parcourt toutes les parties (même si
                          % pour les lettres que nous choisirons il
                          % n'y en aura qu'une)
    contourLettre := pathpart item;
    for i:=1 upto nbrPoints:
      if i=1:
        p := point i/nbrPoints along contourLettre; % on stocke
                                                    % dans un path tous les points
      else:
        p:= p--(point i/nbrPoints along contourLettre);
      fi;
    endfor;
    draw p scaled scale;
  endfor;
endfig;
end;

```

Ce code produit le résultat suivant (obtenu directement avec l'extension `luampLib` [4]).



Nous n'allons pas détailler ce code ici. Il paraît un peu complexe mais cela est dû au fait que la structure de `glyph` de METAPOST permet (heureusement) d'avoir plusieurs parties pour un glyphe. Même si, ici, nous ne considérerons que des lettres d'une seule partie (connexes), il faut bien se plier à la structure de données. Ce qu'il faut retenir, c'est que nous disposons d'un code METAPOST qui permet d'obtenir un ensemble de points constituant une discrétisation du contour du glyphe du caractère.

La liste de points avec Lua

Le calcul de la décomposition en série de Fourier de cette courbe fermée doit être possible en T_EX, cependant Lua nous offre plus de capacités, plus de rapidité, et plus de facilité de codage⁴. On veut donc récupérer cette liste de points du côté de Lua. L'avantage aussi, c'est que comme nous avons déjà dit, LuaT_EX inclut METAPOST via la librairie Lua, Mplib (voir [6]).

4. Au moins pour moi!

Chercher dans les fichiers de fontes.

Avec METAPOST (ou Mplib), on ne peut malheureusement utiliser qu'une fonte Type1. Il y a une petite subtilité concernant l'ouverture du fichier de fonte. En effet, METAPOST demande un type de fichier pfb, alors que kpse demande un type de fichiers type1 fonts. Taco HOEKWATER, sur la liste de diffusion metapost@tug.org m'a fourni la fonction de recherche qui gère ce petit souci et que je fournis ci-dessous.

```

local mpkpse = kpse.new('luatex', 'mpost')

local function finder(name, mode, ftype)
  if mode == "w" then
    return name
  else
    if ftype == 'pfb' then
      ftype='type1 fonts'
    end
    return mpkpse:find_file(name, ftype)
  end
end
end

```

Depuis METAPOST vers un tableau Lua

Passée cette petite difficulté technique, nous allons ici présenter une fonction Lua qui permet de construire un tableau Lua qui contient les points générés par METAPOST.

```

function getpathfrommp(s, nbrPoints, scale)
  -- on definit une fonction qui récupère la liste de
  -- nbrPoints points faite à partir du contour du caractere s
  -- scale est un paramètre d'homothétie

  --on lance une session MetaPost avec la fonction de recherche
  local mp = mplib.new({find_file = finder,})
  -- on charge le format metafun
  mp:execute('input metafun ;')
  -- on stocke la sortie de l'exécution du code MetaPost
  local rettable
  rettable = mp:execute('fontmapfile "=lm-ec.map"; picture lettre;
  path contourLettre; lettre := glyph "' .. s .. '" of "ec-lmri10";
  path p; beginfig(1);
  for item within lettre: contourLettre := pathpart item; for i:=1
  upto '.. nbrPoints .. ':
  if i=1: p := point i/' .. nbrPoints..' along contourLettre;
  else: p:= p--(point i/' .. nbrPoints..' along contourLettre);fi;
  endfor;
  draw p scaled '..scale..';
  endfor; endfig;end;') -- Code MetaPost présenté plus haut
  -- initialisation du tableau de points de sortie
  output = {}
  -- si l'exécution du code MetaPost s'est bien passée
  if rettable.status == 0 then
    figures = rettable.fig -- la liste des figures
  end
end

```

```

figure = figures[1] -- la première figure (la seule)
local objects = figure:objects() -- la liste des objets
                                -- composant la figure
local segment = objects[1] -- le premier objet (le seul)
for point = 1, #segment.path do
    output[point] = {}
    output[point].x = segment.path[point].x_coord
    output[point].y = segment.path[point].y_coord
end
else
    print("error")
end
return output
end

```

Pour expliquer grossièrement le code ci-dessus, il s'agit de récupérer la sortie de l'exécution d'un code par METAPOST (Mplib ici). Cette sortie a une structure en Lua (il faut explorer la documentation de LuaTeX, section Mplib [6]). On parcourt donc cette structure pour extraire la liste de points qui nous intéressent : tout d'abord la liste de figures, qui ici est limitée à une seule figure, puis à l'intérieur de la première figure, on va chercher les « objects » qui là encore sont limités à un seul « object » (notre courbe fermée), puis on parcourt le « path » (METAPOST) de l'objet, nommé ici « segment », et enfin on récupère les coordonnées x et y qu'on stocke dans notre variable de sortie « output ». Pour avoir accès aux fonctions Lua permettant de parcourir la structure de l'objet produit par l'exécution du code METAPOST, il faut se référer à la documentation de LuaTeX.

On notera qu'on a passé les paramètres du caractère à traiter, du nombre de points, et d'homothétie en arguments de la fonction Lua, le caractère dont on veut tracer le contour étant l'argument s . Notons aussi que ce code n'est pas robuste car si le glyphe correspondant au caractère s n'est pas connexe alors il y a de fortes chances que le code ne fonctionne pas.

La décomposition en séries de Fourier avec Lua

Appel à une bibliothèque extérieure

La décomposition en série de Fourier se fait avec des nombres complexes comme nous l'avons présenté précédemment. Les nombres complexes ne sont pas nativement gérés par Lua, mais il existe de nombreux codes disponibles sur le Web où les fonctions de calcul sur les nombres complexes sont implémentées. J'ai choisi le fichier `complex.lua` disponible à cette adresse :

<http://lua-users.org/wiki/ComplexNumbers>

Pour charger ce code, il faut :

1. du côté de \LaTeX , charger l'extension `luapackageloader` (voir à la fin de cet article pour le code \LaTeX complet);
2. du côté de Lua, appeler le fichier `complex.lua` par la ligne

```
complex = require "complex"
```

Convertir la liste de coordonnées de \mathbb{R}^2 en une liste de nombres complexes.

Pour pouvoir programmer les calculs facilement, on crée une fonction de conversion de la liste de coordonnées obtenue par la fonction Lua `getpathfrommp` en une liste de nombres complexes. Ceci se fait par la fonction suivante qui ne nécessite pas plus d'explications.

```
function pathToComplex(path)
  local complexPath
  complexPath = {}
  for i=1,#path do
    complexPath[i] = complex.new(path[i].x,path[i].y)
  end
  return complexPath
end
```

Implémentation de calcul des coefficients de Fourier.

Le calcul des coefficients de Fourier $\tilde{c}_n(f)$ de l'égalité (2) s'implémente très facilement avec Lua, comme le montre le code suivant.

```
function cn(f,n)
  local CN
  CN = complex.new(0.0,0.0)
  local N = #f
  for i=0,N-1 do
    exposant = complex.new(0.0,-2.0*math.pi*n*i/N)
    Exp = complex.exp(exposant)
    CN = complex.add(CN,complex.mulnum(complex.mul(f[i+1],Exp),1.0/N))
  end
  return CN
end
```

À partir de cela, il nous faut construire la liste

$$(c_{-M/2}, c_{-M/2+1}, \dots, c_{-1}, c_0, c_1, \dots, c_{M/2}),$$

ce qui se fait par la fonction Lua suivante :

```
function cnList(f)
  local CNlist = {}
  local M = #f
  for i=0,M do
    CNlist[i] = cn(f,math.floor(i-M/2))
  end
  return CNlist
end
```

Tracé avec mplibcode

Une fois toutes ces briques de codes préparées, il nous suffit d'*implémenter le tracé* avec l'aide de l'environnement `mplibcode` de l'extension `luampplib`⁵ [4]. Cette fonction a plusieurs arguments :

- un chemin discrétisé (`path`), c'est-à-dire l'ensemble des coordonnées (x, y) du contour du glyphe;
- sa conversion en complexes (`complexPath`);
- une liste des coefficients de Fourier (`cnList`);
- un nombre de coefficients de Fourier que l'on souhaite (`M` dans les équations précédentes), c'est-à-dire le nombre de cercles et vecteurs tracés (`nbrFourier`);
- un temps $t \in [0, 1]$.

```
function coreDecomp(path, complexPath, cnList, nbrFourier, t)
-- path : liste de points de R^2
-- complexePath : conversion de cette liste en complexes
-- cnList : liste des coefficients de Fourier
-- nbrFourier : nombre de coefficients M+1 dans les équations
--initialisations
local str
local cnListRotated = {}
local zero = math.floor(#cnList/2)
local NFourier = math.floor(nbrFourier/2)
cnListRotated[zero] = cnList[zero]
-- on multiplie les coefficients par l'exponentielle e^{2i k pi t}
for k=1, zero do
  cnListRotated[zero+k] = complex.mul(cnList[zero+k], complex.exp(
    complex.new(0.0, k*2*math.pi*t)))
  cnListRotated[zero-k] = complex.mul(cnList[zero-k], complex.exp(
    complex.new(0.0, -k*2*math.pi*t)))
end
-- on commence le code mplibcode
local str = "\\begin{mplibcode}\\nverbatimtex \\leavevmode etex;
beginfig(1);"
local mpCodeLetter = mpCodePath(path) -- code metapost du tracé du
glyphe
-- appel fonction extérieure
str = str .. mpCodeLetter -- concaténation de chaînes de caractères
local currentC = complex.new(0,0) -- le point complexe courant (sur
-- lequel on va placer le cercle
-- suivant)
-- on ajoute le code de tracé du cercle et du vecteur au point courant
str = str .. mpCodeCircle(cnListRotated[zero], currentC)
currentC = complex.add(currentC, cnListRotated[zero])
for i=1, NFourier do -- on itère le procédé pour tous les modes de
Fourier
  str = str .. mpCodeCircle(cnListRotated[zero+i], currentC)
  currentC = complex.add(currentC, cnListRotated[zero+i])
  str = str .. mpCodeCircle(cnListRotated[zero-i], currentC)
end
```

5. Mais nous aurions pu générer du code Tikz par exemple.

```

    currentC = complex.add(currentC, cnListRotated[zero-i])
  end
  str = str .. "endfig;\n\\end{mplibcode}\n\\newpage" -- on ferme tout
  return str
end

```

Pour aider à la lecture du code, notons que `cnListRotated[i]` correspond au termes dans la somme (1) : $\tilde{c}_n(f)e^{2ik\pi t}$, puisque la multiplication par $e^{2ik\pi t}$ peut être vue comme une rotation dans le plan complexe.

Cette fonction a pour but principal de construire une chaîne de caractères qui contient le code `mplibcode` qui sera *envoyé* à \TeX via la fonction `Lua tex.sprint()`.

La fonction `coreDecomp` ci-dessus fait appel à deux autres fonctions Lua qui produisent le code METAPOST du tracé du graphique :

- la fonction `mpCodePath(path)` qui prend en argument la liste des points du contour du glyphe et qui trace ce dernier⁶ ;
- la fonction `mpCodeCircle(cn, shift)` qui prend en argument un coefficient de Fourier `cn` (après rotation) et un couple de \mathbf{R}^2 `shift` qui est l'extrémité de la ligne brisée au bout de laquelle le vecteur et le cercle doivent être dessinés.

Le code de ces deux fonctions est à lire ci-dessous. Elles consistent principalement à la concaténation de chaînes de caractères pour produire du code METAPOST.

```

function mpCodePath(path)
-- trace juste le chemin contour du glyphe en reliant simplement les
-- points entre eux
  local str
  str = ""
  str = str .. "path p;\n"
  str = str .. " p:="
  for i=1,#path do
    str = str .. "(" .. string.format("%f", path[i].x) .. "," ..
      .. string.format("%f", path[i].y) .. ")--"
  end
  str = str .. "cycle;\n"
  str = str .. "draw p;\n"
  str = str .. "pair ll, lr, ur, ul; ll := llcorner p; ur := urcorner
    p; lr := lrcorner p; ul := ulcorner p;\n"
  str = str .. "Width := abs(xpart lr - xpart ll); Hght := abs(ypart ul
    - ypart ll); prcW := 0.8; prcH := 0.3;\n"
  str = str .. "draw (ll+(-prcW*Width, -prcH*Hght))--(lr+(+prcW*Width, -
    prcH*Hght))--(ur+(+prcW*Width, +prcH*Hght))--(ul+(-prcW*Width, +
    prcH*Hght))--cycle;\n"
  return str
end

function mpCodeCircle(cn, shift)
-- trace le cercle et le vecteur correspondant au coefficient de
-- Fourier cn (module et argument), centré au point de R^2 shift
  local str

```

6. Il y a un cadre tracé autour aussi pour faire en sorte que toutes les images aient la même dimension et ainsi pouvoir enchaîner les images pour produire une animation.

```

local abs, arg
abs, arg = complex.polar(cn)
str = "draw fullcircle scaled " .. string.format("%f", 2*abs) ..
      shifted (" .. string.format("%f", shift[1]) .. ", " .. string.format("
      %f", shift[2]) .. ") withcolor (0.7,0.7,0.7);\n"
str = str .. "drawarrow ((0,0)--(" .. string.format("%f", cn[1]) .. ", "
      .. string.format("%f", cn[2]) .. ") ) shifted (" .. string.format("
      %f", shift[1]) .. ", " .. string.format("%f", shift[2]) .. ") withpen
      pencircle scaled 1pt withcolor (0.7,0.3,0.3);"
return str
end

```

Générer les images pour tout $t \in [0, 1]$

Pour créer l'animation, il suffit de générer les images avec une discrétisation de l'intervalle de temps $[0, 1]$. Cela peut se faire avec la fonction suivante.

```

function plotDecompAnim(letter, nbrPoints, nbrFourier, scale, nbrFrame)
-- letter : caractère dont on veut faire la décomposition du glyphe
-- nbrPoints : nombre de points pour la discrétisation du contour
-- nbrFourier : nombre de modes pour la décomposition de Fourier
-- scale : facteur d'homothétie
-- nbrFrame : nombre pour diviser l'intervalle [0,1], et donc nombre d'
  images
  local str
  local path = getpathfrommp(letter, nbrPoints, scale)
  local complexPath = pathToComplex(path)
  local cnList = cnList(complexPath)
  local cnListRotated = {}
  local zero = math.floor(#cnList/2)
  local NFourier = math.floor(nbrFourier/2)
  for frame=0, nbrFrame-1 do
    t=frame/nbrFrame
    str = coreDecomp(path, complexPath, cnList, nbrFourier, t)
    tex.sprint(str)
  end
end

```

Cette fonction fait appel aux fonctions précédemment présentées ainsi qu'à la fonction de LuaTeX `tex.sprint()` qui permet d'envoyer la chaîne de caractères à \TeX pour la composition.

Les fonctions Lua présentées sont toutes mises dans un fichier unique `Fourier.lua`.

Animations et code

Une fois toutes ces fonctions Lua implémentées, il suffit de faire appel à la fonction Lua `plotDecompAnim` grâce à la macro `\directlua` comme le montre le code suivant.

```
\documentclass{article}
\usepackage{luapackageloader}
\usepackage{luamplib}

\directlua{dofile("Fourier.lua")}
\pagestyle{empty}

\begin{document}
\directlua{
  plotDecompAnim("f",300,50,0.26,360)
}
\end{document}
```

Pour conclure, nous n'avons considéré que 3 fichiers, le fichier \LaTeX ci-dessous `fourier.tex`, le fichier `Fourier.lua` qui contient toutes nos fonctions Lua présentées ici, et le fichier `complex.lua` récupéré sur le web. Pour compiler et produire le PDF qui contient autant de pages que d'images, il suffit de mettre ces trois fichiers dans le même répertoire et d'exécuter `lualatex` sur notre fichier `fourier.tex`.

```
user $> lualatex fourier.tex
```

Si vous lisez ce PDF avec Acrobat Reader, vous pourrez lire l'animation générée (cf. figure 2) avec l'extension `animate` [1].

Sinon, tous les codes et l'animation sont visibles et téléchargeables à l'adresse suivante :

<https://fougeriens.org/~mc/?page=exemples&dir=fourier>

Maxime CHUPIN

Références

- [1] G. ALEXANDER, *The animate package*, 2020 <https://ctan.org/pkg/animate>,
- [2] ASSOCIATION GUTENBERG, *Cahiers n° 54-55 : Introduction à Lua \TeX* , Octobre 2010, http://cahiers.gutenberg.eu.org/cg-bin/feuilleter?id=CG_2010__54-55.
- [3] HANS HAGEN, *Metafun*, 2017, <http://www.pragma-ade.com/general/manuals/metafun-p.pdf>, v 2.0
- [4] HANS HAGEN, TACO HOEKWATER, ELIE ROUX, PHILIPP GESANG ET KIM DOHYUN, *The luamplib package*, 2020, <https://ctan.org/pkg/luamplib>, v. 2.11.3.
- [5] JOHN D. HOBBY AND THE METAPOST DEVELOPMENT TEAM, *Metapost, a user's manual*, 2019, <https://ctan.org/pkg/metapost>, v. 2.0.
- [6] L $\text{UA}\text{T}\text{E}\text{X}$ DEVELOPMENT TEAM, *Lua \TeX reference manual*, March 2020, dernière version en ligne v. 1.12 : <http://www.luatex.org/svn/trunk/manual/luatex.pdf>.
- [7] ROBERTO IERUSALIMSCHY, *Programming in Lua*, 2016, <https://www.lua.org/>.

FIGURE 2 – Animation générée

CAMPUS DU LIBRE, ÉDITION 2020



Initialement prévue sous forme de stand, notre participation à cette troisième édition du *Campus du Libre* a été transformée en démonstration à distance, pour tenir compte des mesures sanitaires. Le samedi 7 novembre 2020, nous avons donc présenté en visioconférence \LaTeX , GUTenberg, ainsi que nos projets récents, à une trentaine de personnes, avant de faire une démonstration de LearnLaTeX.org et de notre FAQ.

<http://www.campus-du-libre.org/>

Un rapide sondage en début de présentation a montré que 60 % de l'auditoire ne connaissait pas \LaTeX , ou seulement de nom. La discussion s'est poursuivie pendant près de 50 minutes après la fin de l'exposé, avec des questions de curiosité, des questions techniques, et beaucoup d'enthousiasme!

Au final, les échanges ont été aussi ouverts que sur un stand physique, et les participants avaient en plus la possibilité d'assister à la présentation sans forcément s'engager dans une discussion. Un grand nombre a été consulter les sites web évoqués, pendant et après la présentation. Nous avons donc trouvé beaucoup d'avantages à cette formule, que nous essaierons de reproduire en complément de notre présence matérielle sur des salons à venir!

Jérémy JUST

COMPTE RENDU DE LECTURE



Christian TELLECHEA, *Apprendre à programmer en TÈX*, Septembre 2014, CTAN, <https://ctan.org/pkg/apprendre-a-programmer-en-tex>, 580 pages.

En avril 2019, Christian TELLECHEA, bien connu sous le pseudonyme *unbonpetit* dans les divers groupes de discussion sur TÈX, a déposé son livre *Apprendre à programmer en TÈX* sur le CTAN. Il s'agit d'un ouvrage auto-édité dont l'impression à la demande n'est malheureusement plus possible, mais qui est devenu accessible à tous sur le CTAN.

Malgré la déception de ne plus pouvoir acquérir ce livre, il faut remercier l'auteur de le rendre disponible car il s'agit d'un livre remarquable à bien des égards. La programmation en TÈX n'est pas quelque chose de facile et ce livre permet de s'initier à cette science qui s'approche souvent de la magie noire pour beaucoup d'utilisateurs et d'utilisatrices. L'auteur fait preuve d'une pédagogie rare nous amenant progressivement à comprendre les arcanes de TÈX. Toutes les explications sont accompagnées de nombreux exemples (plus de 450 codes commentés), toujours très bien choisis. Christian nous propose aussi quelque 250 exercices qui nous permettent de ne pas en rester à une lecture superficielle, mais bien de mettre en pratique, et ainsi de vraiment progresser. De plus, les notions sont régulièrement illustrées avec soin, par exemple le délicat mécanisme de développement en TÈX, ce qui aide grandement la compréhension.

Même si l'auteur nous avertit bien que cet ouvrage n'est qu'une *introduction*, l'étendue des thèmes abordés est considérable et, selon nous, fait de cet ouvrage une référence pour la programmation en TÈX qui mériterait une traduction vers l'anglais, au moins. Cela fait que ce livre peut aussi être utile aux moins débutants cherchant quelques compléments ou éclaircissements sur telle ou telle notion de programmation TÈXienne. Bref, vous l'aurez compris, nous vous conseillons vivement de vous jeter dans la lecture et l'étude de cet ouvrage!

Maxime CHUPIN

GUTenberg

Association GUTenberg
c/o Cellule MathDoc – U.M.S. 5638 – BP 74
38402 Saint-Martin-d'Hères Cedex, France
secretariat@gutenberg.eu.org

Site GUTenberg : <https://www.gutenberg.eu.org/>

Cahiers GUTenberg : <http://cahiers.gutenberg.eu.org/>

Problèmes techniques :

la liste gut : <https://www.gutenberg.eu.org/-Listes-de-diffusion->

le site TeXnique de questions et réponses : <https://texnique.fr/>

la FAQ : <https://faq.gutenberg.eu.org>

**Cette association est la vôtre : faites-nous part de vos idées, de vos
envies, de vos préoccupations à l'adresse
secretariat@gutenberg.eu.org.**

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 2021

Les membres de GUTenberg peuvent adhérer au TUG (*T_EX User Group*) et recevoir son bulletin *TUGboat* à un tarif préférentiel :

tarif normal : 60 € (au lieu de 85\$)

tarif étudiant : 40 € (au lieu de 55\$)

Voir <https://www.gutenberg.eu.org/?Adherer-a-l-association>

Type d'adhésion	Prix
Membre individuel	30 €
Membre individuel + adhésion TUG	90 €
Membre individuel étudiant/demandeur d'emploi	15 €
Membre individuel étudiant + adhésion TUG	55 €
Association d'étudiants	65 €
Organisme à but non lucratif	130 €
Organisme à but lucratif	229 €

Les règlements peuvent s'effectuer :

- en ligne (CB ou Paypal) :
<https://www.gutenberg.eu.org/?Adherer-en-ligne>
- de façon traditionnelle (bulletin et chèque) :
<https://www.gutenberg.eu.org/?Adherer-a-l-association>

La Lettre GUTenberg

Directeur de la publication : Patrick BIDEAULT

Adresse de la rédaction : GUTenberg

Cellule Mathdoc

B.P. 74, F-38402 St-Martin-d'Hères Cedex (France)

ISSN : 2742-6149 (version numérique)