

CAHIERS

GUTenberg

☞ DONALD KNUTH : DES MATHÉMATIQUES À LA TYPOGRAPHIE

☞ Sandrine CHENEVEZ, Myriam HAMLÀ, Stéphanie LOUISON

Cahiers GUTenberg, n°58 (2021), pages 5–33.

<https://doi.org/10.60028/cahiers.v2021i58.31>

© Association GUTenberg, 2021, tous droits réservés.

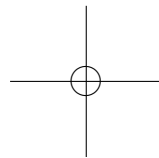
L'accès aux articles des *Cahiers GUTenberg* :

<https://publications.gutenberg-asso.fr/cahiers>

implique l'accord avec les conditions générales d'utilisation :

<https://publications.gutenberg-asso.fr/cahiers/about>

Toute utilisation commerciale ou impression systématique est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de *copyright*.



☞ DONALD KNUTH : DES MATHÉMATIQUES À LA TYPOGRAPHIE¹

☞ Sandrine CHENEVEZ, Myriam HAMLÀ, Stéphanie LOUISON

RÉSUMÉ. Cet article retrace les principales étapes de la vie de Donald Ervin Knuth, puis ses apports à l'informatique et à la typographie, principalement à travers sa série d'ouvrages *The Art of Computer Programming* et le duo $\text{T}_{\text{E}}\text{X}/\text{M}\text{E}\text{T}\text{A}\text{F}\text{O}\text{N}\text{T}$. Sa lecture ne requiert pas de connaissances techniques particulières.

ABSTRACT. *This article traces first the main stages of Donald Ervin Knuth's life, then his contributions to computer science and typography, mainly through its book series The Art of Computer Programming and the $\text{T}_{\text{E}}\text{X}/\text{M}\text{E}\text{T}\text{A}\text{F}\text{O}\text{N}\text{T}$ pair. Reading it does not require special technical knowledge.*

ZUSAMMENFASSUNG. *Dieser Artikel beschreibt die wesentlichen Punkte im Lebens von Donald Ervin Knuth, beschäftigt sich mit seinen Beiträgen zur Informatik und Typografie hauptsächlich anhand seiner Buchreihe The Art of Computer Programming und des $\text{T}_{\text{E}}\text{X}/\text{M}\text{E}\text{T}\text{A}\text{F}\text{O}\text{N}\text{T}$ Duo. Zum Lesen dieses Artikels sind keine besonderen technischen Kenntnisse nötig.*

NOTE. Ce travail a été effectué à l'automne 2020 dans le cadre de l'UE² « Histoire de l'édition, des métiers et des techniques connexes », enseignée par Éric GUICHARD dans le *master* « Publication numérique » de l'ENSSIB³. Quelques corrections éditoriales ont été apportées par Éric GUICHARD et Jean-Michel HUFFLEN, qui remercient Emmanuel SAINT-JAMES, Denis B. ROEGEL et Yvon HENEL pour leurs précieux commentaires et suggestions.

1. Titre en anglais : *Donald Knuth: From Mathematics to Typography*; en allemand : *Donald Knuth: von der Mathematik zu Typografie*.

2. Unité d'Enseignement.

3. École Nationale Supérieure des Sciences de l'Information et des Bibliothèques.

INTRODUCTION

DONALD ERVIN KNUTH est un informaticien, mathématicien et typographe américain de renom. Il est un des pionniers de l'algorithmique et a apporté de nombreuses contributions à plusieurs branches de l'informatique théorique. Il est l'auteur d'une centaine d'articles et d'une dizaine de livres sur les mathématiques et l'algorithmique. Son œuvre majeure, composée actuellement de quatre imposants volumes et non terminée à ce jour, est *The Art of Computer Programming*, dans laquelle il fusionne sa double passion pour les problèmes numériques discrets et les grandes collections d'informations.

En 1968, il devient membre de la faculté de l'Université de Stanford, où il recevra plus tard un titre académique créé spécialement à son intention : *Professor Emeritus of the Art of Computer Programming*. En 1971, Knuth est le premier à recevoir le prix *ACM Grace Murray Hopper Award*. Il a reçu de nombreuses autres distinctions honorifiques, entre autres le prix Turing, la *National Medal of Science* (États-Unis), la médaille John Von Neumann de l'IEEE⁴, ainsi que le prix de Kyoto et la Médaille Franklin. Il est élu membre associé de l'Académie des sciences française en 1992 et membre de la *Royal Society* en 2003.

Afin d'avoir une bonne qualité de mise en page pour la deuxième édition du deuxième volume de son *opus* « *The Art of Computer Programming* », Knuth crée deux logiciels libres, par la suite largement utilisés en typographie professionnelle et en mathématiques, $\text{T}_{\text{E}}\text{X}$ et METAFONT. Son intérêt pour la typographie le pousse également à créer la police *Computer Modern*, police par défaut de $\text{T}_{\text{E}}\text{X}$.

Âgé de plus de quatre-vingts ans (il est né en 1938), il vit aujourd'hui à Stanford, au sud de San Francisco, dans la maison californienne moderne que lui et sa femme Jill ont fait construire en 1970. Son bureau est jonché de piles de clés USB, et le plus impressionnant est la salle de musique, construite autour de son orgue à 812 tuyaux sur mesure (ROBERTS 2018).

Nous retraçons les principales étapes de sa vie dans le § 1. Puis la genèse et l'avancement de son œuvre la plus imposante en matière de publication scientifique est abordée dans le § 2. Quant au cheminement vers $\text{T}_{\text{E}}\text{X}$ et METAFONT, il est décrit dans le § 3. Quelques réalisations plus marginales

4. *Institute of Electrical and Electronics Engineers*.



FIGURE 1. Donald Knuth en 2010.

sont abordées dans le § 4. Pour les ouvrages dont il existe plusieurs éditions, la première citation se place à l'époque de la première édition, mais c'est l'édition la plus récente qui est spécifiée dans la section bibliographique de l'article.

I. LE PERSONNAGE : UN GÉNIE PRÉCOCE

1.1. FAMILLE ET ENFANCE (1938–1956)

Donald Ervin Knuth est né le 10 janvier 1938 à Milwaukee, dans le Wisconsin (Nord-Est des États-Unis). Ses parents étaient venus s'y installer avant sa naissance afin que son père puisse y prendre un poste d'enseignant dans une école luthérienne. Ils ne sont pas aisés financièrement, l'emploi de son père étant très mal rémunéré. Aussi, ce dernier doit cumuler d'autres petits emplois en complément de son métier dans l'éducation ; il achète également des actions, notamment dans la *Walt Disney Company*. Cependant, c'est surtout l'emploi de sa mère, travaillant initialement en

tant que secrétaire tout en continuant à s'occuper de sa maison et de sa famille, puis promue à de plus hautes fonctions au fil des années, qui les mettra à l'abri du besoin (KNUTH 2020c). Ainsi, ses parents, venant de familles peu éduquées, sont des précurseurs puisqu'ils travaillent tous les deux. Ils souhaitent la meilleure éducation pour leur fils. Ils commencent à le familiariser avec la lecture dès son plus jeune âge : à deux ans et demi, ils l'emmènent déjà à la bibliothèque (*ibid.*)

Donald Knuth est très précoce, comme le montre une anecdote de son enfance. Alors qu'il a treize ans, une entreprise de confiserie organise un concours : déterminer combien de mots pourraient être faits à partir des lettres du nom du bonbon *Ziegler's Giant Bar*. Le jeune Donald feuillette alors méthodiquement le dictionnaire de sa famille (de l'éditeur Funk & Wagnalls⁵) composé de pas moins de 2 000 pages. Il étiquette les sections de l'ouvrage avec des titres tels que « Aa », « Ab » et « Ba » en fonction des débuts de mots possibles en utilisant des lettres du nom du bonbon, puis il parcourt les colonnes du dictionnaire en notant les mots qui se qualifient. Il constate qu'il peut ignorer des parties entières du dictionnaire, telles que des pages contenant des mots commençant par la lettre « C » ou des parties des mots commençant par « B » et dont la deuxième lettre est « U ». Finalement, Knuth relève plus de 4 700 mots, alors que les responsables du concours eux-mêmes n'en ont identifié qu'environ 2 000. Il a déjà un esprit des plus logiques, et une appétence pour la résolution de problèmes. Il se définit lui-même comme un *geek*⁶ dès cette époque. Il dit d'ailleurs, dans une interview qu'il a accordée à la revue *Quanta Magazine* : « J'ai eu une tendance obsessionnelle-compulsive qui m'a attiré vers des problèmes numériques discrets. Et j'ai adoré examiner de grandes collections d'informations » (D'AGOSTINO 2020).

Ses prédispositions amènent Donald Knuth à faire une brillante scolarité dans les écoles luthériennes qu'il fréquente : au lycée, il décroche un record avec une moyenne de 97,5/100. Malgré ses études plus qu'honorablement réussies, Knuth souffre d'un complexe d'infériorité : contrairement à certains de ses camarades, il n'a jamais sauté de classe, est en grande difficulté en sport, et travaille donc dur pour prouver qu'il peut réussir. Il est déjà

5. Voir le lien <http://www.funkandwagnalls.com/>.

6. Personne passionnée par un ou plusieurs domaines, souvent en relation avec l'informatique.

très exigeant : il pense que les leçons de mathématiques qu'il reçoit alors ne sont pas très poussées par manque d'ambition dans les programmes scolaires et par manque de compétences des enseignants. Cependant, il apprécie tout de même les professeurs de ses écoles religieuses, qui sont attentionnés, très attentifs aux besoins de leurs élèves, presque comme s'ils étaient investis d'une mission de transmission du savoir. Il est très intéressé par la physique-chimie — car il admire beaucoup son enseignant dans cette matière — ainsi que par la musique. Cette dernière est une vraie passion pour lui : il apprend le piano, chante dans la chorale de l'école et joue du tuba et du saxophone dans un groupe. Il essaie même de composer ou d'arranger des morceaux. De plus, il trouve ses cours d'anglais très intéressants, et rédige durant son temps libre un journal lycéen avec ses camarades, sa première expérience d'écriture, et même d'édition. En effet, ayant chez lui une machine à écrire et une autre dédiée à la copie, il est chargé tous les lundis soirs de mettre en page, d'imprimer et de reproduire les exemplaires du journal (KNUTH 2020c).

1.2. UNIVERSITÉ (1956–1960)

En 1956, après le lycée luthérien, Knuth entre à l'université, à laquelle seuls 7 à 8 % des élèves accèdent à cette époque. Il choisit d'étudier au *Case Institute of Technology* de Cleveland (aujourd'hui devenu la *Case Western Reserve University*), qu'il pense être l'une des plus prestigieuses universités, car nombreux d'élèves y échouent. Il souhaite donc faire ses preuves là-bas, et en a l'occasion puisqu'il intègre la section d'honneur, une classe d'une vingtaine d'étudiants triés sur le volet en raison de leurs excellents résultats préalables. Il y suit des cours d'anglais, de physique, de chimie et de mathématiques (*ibid.*)

Son enseignant de mathématiques en première année, Paul Guenther, suscite son admiration par ses connaissances pointues dans son domaine, mais également en physique et en chimie. Knuth travaille extrêmement dur dans la matière de Guenther, puis de Louis Green (son professeur les années suivantes), afin d'impressionner ceux-ci, et se met à apprécier les mathématiques. D'autant plus que les sciences comme la physique et la chimie lui déplaisent par les expérimentations qu'elles impliquent ainsi que par l'incertitude qui y règne. Il aime pouvoir démontrer avec certitude qu'un problème a une solution et déterminer laquelle, comme

on peut le faire en mathématiques. Au contraire, la physique et la chimie tentent d'expliquer le monde réel, qu'il estime beaucoup trop complexe pour obtenir des réponses certaines aux questions que l'on se pose. Il dit d'ailleurs, à l'âge de 80 ans, qu'il pense que les êtres humains ne connaissent qu'une partie infinitésimale du monde. C'est en particulier le cours de Louis Green intitulé *Basic Mathematics* qui crée un déclic : il y étudie l'arithmétique (avec des nombres entiers), et non plus des fonctions continues comme c'est le cas en physique. Il se sent bien plus à l'aise avec les nombres entiers et les suites (donc avec les mathématiques discrètes), ce qui l'amène naturellement vers les algorithmes (KNUTH 2020c).

Durant son temps libre à l'université, il pratique de très nombreuses activités, incluant son implication dans la fraternité dont il est membre (et vice-président), la musique et le sport. En particulier, dans le domaine sportif, il cherche à se rendre utile à l'équipe de basketball de son institution. Pour cela, il utilise des formules mathématiques qu'il implémente sur l'ordinateur IBM 650 auquel il a accès au sein de l'université. Le mécanisme est le suivant : il relève des données au cours des matchs (possessions de balle, perte de balle, etc.), puis il estime en termes de probabilités la contribution de chaque joueur de l'équipe lors des matchs. Par exemple, la possession de la balle par un joueur est associée à une probabilité de 7 chances sur 10 pour l'équipe de marquer un point. En revanche, la perte de la balle lui fait perdre des points, alors qu'une reprise de balle peut lui en rapporter. Ainsi, grâce à des probabilités et aux capacités de l'informatique, il aide pendant quatre ans (de 1956 à 1960) le *coach* à organiser et diriger son équipe, et à lui faire remporter sa ligue. Les journaux locaux relaient cette nouvelle méthode d'entraînement incluant des ordinateurs, IBM est intéressé par le sujet qui constitue une bonne publicité, et vient de tourner un reportage dans lequel Knuth apparaît. Ce court film est ensuite diffusé à l'échelle nationale. Il est désormais disponible sur la chaîne YouTube du *Computer History Museum* sous le titre de *The Electronic Coach*⁷ (ROBERTS 2018).

7. La vidéo est accessible *via* le lien :

<https://www.youtube.com/watch?v=dhh8Ao4yweQ>

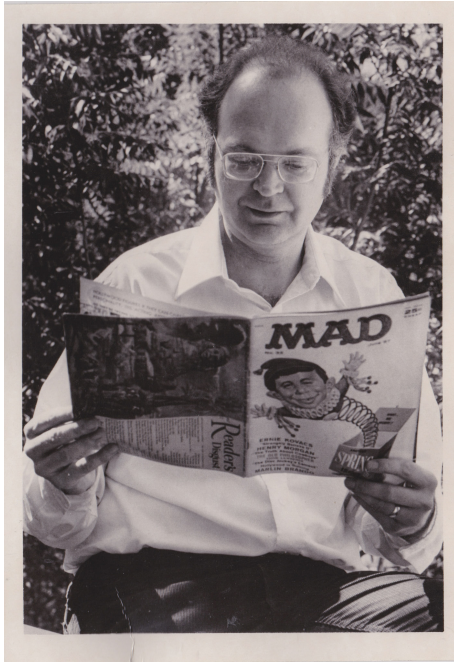


FIGURE 2. Knuth en 1981, regardant le numéro du magazine *Mad* de 1957 qui contenait son premier article technique (KNUTH 1957).

D'autre part, il poursuit à l'université l'expérience du journal de lycée en y éditant un magazine dédié aux sciences qu'il crée avec des collègues : *Engineering and Science Review*. Il apprend alors beaucoup en matière de publication. Par exemple, il a pour rôle de vérifier les coquilles avant l'impression définitive et doit pour cela se rendre sur les lieux d'impression de la revue, dans le centre-ville de Cleveland, où il se familiarise avec les nouvelles techniques d'impression de son époque, notamment la machine linotype. Il considère que, durant ses années universitaires, il a travaillé 50 % de son temps sur les mathématiques, et 50 % sur l'anglais et la rédaction. Il estime que cette mixité entre langage mathématique

et langage naturel est sans doute une caractéristique qui s'est plus tard retrouvée dans tous ses travaux (KNUTH 2020c).

Par ailleurs, Knuth doit travailler pour financer ses années d'université. Il trouve un emploi à temps partiel dans lequel il doit réaliser des graphiques pour les statisticiens du campus. Cela lui plaît beaucoup car il est fasciné par les graphiques depuis le lycée (*ibid.*)

Il est alors installé dans la salle contiguë à celle contenant l'ordinateur loué par la faculté, le modèle 650 d'IBM. Il s'agit du tout premier ordinateur produit en masse avec environ 1 000 exemplaires produits (contre une douzaine maximum pour les précédents). Knuth est fasciné par cet appareil qui occupe une pièce entière, avec toutes ses lumières. Il doit l'utiliser

pour trier des données, mais, voyant son intérêt pour la machine, un informaticien lui explique son fonctionnement et lui montre les cartes sur lesquelles sont inscrits les programmes (KNUTH 2020c).

Il lit le manuel donné par IBM avec des exemples de programmes, et ne les trouve pas satisfaisants. Il les réécrit, avec des solutions plus élégantes. Il réalise alors qu'il a peut-être un talent pour la programmation informatique. Par la suite, les membres du centre informatique le laissent emprunter l'ordinateur durant la nuit pour écrire des programmes qui sont ensuite utilisés par les autres étudiants. Knuth est encore en première année d'université (*ibid.*)

Son premier programme consiste à trouver comment décomposer un nombre en facteurs premiers. L'algorithme consiste alors à diviser le nombre par tous les nombres premiers, les uns après les autres. Il contient alors 130 instructions et Knuth doit faire face à 200 *bugs*. De plus, il faut quatre millisecondes à l'ordinateur pour effectuer un calcul, ce qui rend le processus extrêmement long pour effectuer l'intégralité des calculs. Pour donner un ordre de comparaison, c'est un million de fois plus lent que nos ordinateurs aujourd'hui. Malgré toutes ces difficultés, Knuth ne se décourage pas et continue, au contraire, à manipuler la machine, en cherchant des solutions à tous les problèmes, et en simplifiant son algorithme (par exemple, en limitant le nombre de diviseurs possibles aux nombres inférieurs ou égaux à la racine carrée du nombre, ce qui supprime beaucoup plus de la moitié des calculs).

Il poursuit son apprentissage autodidacte de l'ordinateur pendant ses vacances d'été en lisant des algorithmes de tris, écrits non plus dans le langage binaire de l'IBM 650 mais dans sa version symbolisée, le langage d'assemblage SOAP⁸ II créé par Stan Poley de l'entreprise IBM, puis dans un des tout premiers langages évolués, IT⁹, écrit par Alan Perlis et des collègues de l'université de Carnegie, qui traduit en binaire des formules algébriques. Autrement dit, au lieu de devoir rédiger un programme sur carte avec toutes les valeurs à tester listées, Knuth découvre qu'il est possible de programmer avec des variables prenant la forme de lettres, auxquelles sont ensuite affectées diverses valeurs numériques. Par exemple, pour effectuer des sommes, au lieu d'écrire une suite de chiffres en langage machine, il

8. *Symbolic Optimum Assembly Program*.

9. *Internal Translator*.

peut noter :

$$X1, Z, X2, S, X3$$

Cela signifie que la variable $X1$ prend la valeur égale à la somme des valeurs affectées aux variables $X2$ et $X3$ (Z correspond au signe $=$ qui n'existe pas encore en langage machine; de même, S correspond au signe $+$, non encore disponible sur machine à l'époque). C'est une avancée non négligeable pour la rédaction de programmes qui sont plus parlants pour les mathématiciens, qui ont l'habitude de ce genre d'abstractions (*ibid.*)

C'est d'ailleurs avec ces notations algébriques que Knuth crée à l'été 1957 un programme de jeu de morpion, pour lequel il trouve plusieurs solutions. En particulier, l'une d'entre elles consiste à créer un programme expert, qui connaît déjà les coups intéressants à jouer. Un autre, au contraire, ne sait pas comment jouer et apprend de ses erreurs : il évalue chaque coup joué sur 10 selon qu'il gagne ou qu'il perde. Chaque coup est initialisé à 4. Si le programme débutant perd, il enlève un point aux coups qu'il a joués, en revanche s'il gagne, il ajoute un point. On retrouve ici l'idée d'évaluer la contribution de chaque mouvement à l'ensemble du jeu, que Knuth utilise aussi en sport comme nous l'avons déjà vu. Finalement, le programme expert et le programme débutant fonctionnent bien, et Knuth a même l'idée de les faire jouer l'un contre l'autre pour que le programme expert entraîne le programme débutant. Après 120 parties, le programme débutant ne perd plus face au programme expert, ce qui prouve qu'il a réussi à apprendre. Knuth a réussi ce programme très complexe en un mois, ce qu'il trouve long (*ibid.*)

Cependant, les langages de programmation de l'époque ne lui paraissent pas assez élégants, et il s'emploie avec quelques amis à rédiger des améliorations au langage `IT`, en tenant compte de la contrainte de la taille mémoire de l'ordinateur IBM 650 (10 kilobits). Cela donne naissance au compilateur `RUNCIBLE`¹⁰. Il s'agit d'un programme qui traduit des formules algébriques en une série d'instructions en langage machine. Il écrit un article sur ce programme `RUNCIBLE` qu'il envoie à la revue scientifique *Communications of the ACM*. Il s'agit de sa première expérience de publication dans une revue scientifique (*ibid.*)

10. *Revised Unified New Compiler Ir Basic Language Extended.*

Très doué pour les études, Knuth reçoit en 1960 son *bachelor's degree* (équivalent américain d'une licence) *summa cum laude* (avec les félicitations du jury) simultanément avec son *master's degree*, une décision sans précédent prise par le jury considérant son travail de BSc¹¹ comme valant un MSc¹² en mathématiques (O'CONNOR et ROBERTSON 2015). Il explique plus tard qu'il avait choisi de participer à des cours de *master* car il les considérait plus faciles que les cours de niveaux inférieurs (KNUTH 2020c).

Il souhaite alors devenir professeur d'université, et faire de la recherche en mathématiques, mais n'envisage pas encore de faire carrière dans l'informatique. Il choisit donc de faire un doctorat (PhD) de mathématiques au *California Institute of Technology*, situé à Pasadena.

1.3. DOCTORAT (1960–1963)

Durant l'été 1960, il cherche un travail pour gagner suffisamment d'argent pour épouser Jill, rencontrée à l'université de Cleveland. Il reçoit une offre de la *Burroughs Corporation*, compagnie créée au XIX^e siècle, mais qui a été l'une des premières à investir dans l'informatique et qui produit à cette époque les ordinateurs qui inondent le système bancaire. Cette société lui demande d'écrire un compilateur pour le langage de programmation ALGOL¹³ qu'elle développe sur les ordinateurs Burroughs 205, moyennant la somme de 5 500 \$. Il y travaille presque exclusivement de façon manuscrite pendant un mois, puis il met au point le système pendant quatre ou cinq mois (*ibid.*)

Ayant apprécié cette expérience et les personnes rencontrées, il continue à travailler pour la *Burroughs Corporation* pendant 8 ans, en tant que consultant (INFORMIT 2020). Il est alors chargé de faire le lien entre les ingénieurs, qui construisent des *hardwares* de plus en plus sophistiqués, et les concepteurs de ces machines. Il est alors en contact avec tous les acteurs de l'informatique, ayant à la fois la vision théorique de la conception et celle de la réalisation pratique des ordinateurs. Il est même chargé d'identifier des erreurs quand les ingénieurs pensent que tout fonctionne,

11. *Bachelor of Science*, grade universitaire américain correspondant en France à la licence.

12. *Master of Science*.

13. *ALGO*rithmic Language.

et il en trouve toujours plusieurs centaines lors d'applications auxquelles les précédents testeurs n'avaient pas pensé (KNUTH 2020c).

En parallèle de cette activité à temps partiel, il étudie les mathématiques pour sa thèse à l'université. Cette dernière, sous la direction du mathématicien Marshall Hall, a pour sujet un problème de géométrie théorique concernant la projection de plans¹⁴. Il la soutient en 1963 (KNUTH 1963) car c'était prévu ainsi, mais il trouve la solution au problème qu'il traite en deux heures grâce à un programme informatique qui a déjà été implémenté par R. J. Walker (travaillant à Princeton). Il généralise ce résultat en un théorème, le montre à son directeur de thèse, qui lui répond : « *Well Don, this is your thesis. Write this up and get out of here* ». Il y retravaille pendant quelques mois car il culpabilise d'avoir fini sa thèse aussi vite, comparativement à ses camarades (KNUTH 2020c).

1.4. PREMIÈRES ANNÉES DE CARRIÈRE (1963–1968)

Après sa thèse, il est engagé en tant que professeur-assistant¹⁵ dans le département de mathématiques du *California Institute of Technology*, tout en continuant à travailler pour la *Burroughs Corporation*. Il participe également aux comités éditoriaux de douze revues techniques différentes (*ibid.*) Il écrit aussi un livre sur les compilateurs, sur lequel nous reviendrons dans le § 2.

D'autre part, il concilie ses capacités pour la programmation, sa passion pour les problèmes à solutions exprimées mathématiquement et son amour pour la grammaire anglaise au travers d'un intérêt croissant pour les langages de programmation (*ibid.*) Il s'agit pour lui d'étudier les langages informatiques avec des outils mathématiques. Il se documente sur les *context-free languages*¹⁶, qui ont des propriétés mathématiques intéressantes. Il réalise plus tard qu'il serait intéressant de disposer d'une analyse à double niveau, en ajoutant des attributs *sémantiques* à des expressions dont la fonction grammaticale a été reconue. De plus, il voit un autre lien entre informatique et mathématiques : en effet, il entretient à l'époque une correspondance avec Bob Floyd, un informaticien, qui lui explique qu'il est possible de démontrer qu'un programme informatique

14. *The problem of symmetric block designs with lambda equals two.*

15. C'est l'équivalent d'un maître de conférences dans une université américaine.

16. Il s'agit de langages dont l'analyse ne nécessite aucune information de contexte.

est correct comme on développerait une preuve mathématique (KNUTH 2020c). Cette découverte lui inspire plus tard la célèbre phrase par laquelle il met en garde les utilisateurs d'un de ses logiciels : « Faites attention aux *bugs* dans ce code ; je n'ai fait que démontrer qu'il était correct, je ne l'ai pas essayé » (O'CONNOR et ROBERTSON 2015). Cette nouvelle vision le conforte dans sa double spécialité, à la fois travaillant en informatique et enseignant les mathématiques.

Sur le plan familial, il devient père et a deux enfants avec sa femme Jill : un garçon, John, né en 1965 et une fille, Jennifer, née en 1966. Il s'épuise à jongler entre ses différentes obligations, et finit à l'hôpital avec un ulcère. Il prend alors du recul sur son existence, décide de prendre plus de temps pour être créatif, de se donner moins de contraintes professionnelles et de travailler dans le domaine qui l'intéresse vraiment : l'analyse d'algorithmes. Cela consiste à évaluer quantitativement la qualité d'un algorithme, en termes de rapidité d'exécution, de nombres d'opérations, etc. Par exemple, il peut calculer qu'un programme présente un gain de 13,8 % en nombre d'opérations, nécessite plus ou moins de mémoire, etc. L'analyse d'algorithmes n'est pas encore un champ de recherche, mais c'est le sujet autour duquel tournent finalement tous ses projets depuis quelques années, en particulier la rédaction de son *opus* phare : *The Art of Computer Programming* (KNUTH 2020c).

1.5. CARRIÈRE UNIVERSITAIRE DANS L'INFORMATIQUE

L'Université de Stanford crée en 1965 un département dédié à l'informatique. C'est l'une des premières à se lancer dans cette discipline car les ordinateurs sont, à l'époque, considérés comme des moyens d'effectuer des opérations pour d'autres sciences, telles que la physique ou la biologie. Peu de gens perçoivent alors, dans les milieux universitaires, que l'informatique puisse devenir un objet d'étude à part entière. Cependant, George Forsythe, lui, voit le potentiel de la discipline et ouvre une section dédiée à Stanford, non loin de San Francisco. Il regroupe autant de spécialistes de l'informatique que possible autour de ce projet, et Knuth est ainsi recruté en 1968 pour y devenir professeur (*ibid.*)

Il n'entre pourtant pas en fonction immédiatement, car la guerre du Vietnam l'oblige à donner un an à l'État. Il travaille alors à Princeton dans un service de codage et de décodage de messages. L'année suivante, il s'installe avec son épouse et ses enfants à Stanford, où il fait construire

une maison atypique — comprenant une très grande pièce dédiée à un orgue dont Donald Knuth joue, comme son père avant lui — sur le campus même (*ibid.*)

Il travaille à l'Université de Stanford, jusqu'en 1990, date à laquelle il prend un congé pour se consacrer à l'écriture des derniers volumes de sa série *The Art of Computer Programming*, car il craint que son emploi d'enseignant ne l'empêche de terminer ses livres. Voyant qu'il lui reste encore beaucoup de travail sur ce sujet, il prend sa retraite anticipée en 1993 pour passer le reste de sa vie à terminer l'ensemble de sept volumes (D'AGOSTINO 2020).

2. THE ART OF COMPUTER PROGRAMMING

2.1. LA COMMANDE

En 1962, la maison d'édition Addison-Wesley contacte Knuth et lui demande d'écrire un ouvrage sur les compilateurs. Knuth, qui aime écrire depuis longtemps, et qui a repéré l'éditeur Addison-Wesley pour ses bons manuels scolaires de mathématiques, se lance dans ce projet. Il écrit une table des matières en douze points, considérant que l'on ne peut pas écrire sur les compilateurs sans expliquer toute la programmation. La maison d'édition est enthousiaste et encourage Knuth à poursuivre son projet, bien que celui-ci soit de plus en plus dense (KNUTH 2020c).

De 1962 à 1965, Knuth rédige 3 000 pages manuscrites décrivant l'état des connaissances dans le domaine de l'informatique durant ces années. Puis il tape le premier chapitre à la machine à écrire, l'envoie à son éditeur. Celui-ci réalise que le livre sera beaucoup trop long et donc beaucoup trop cher pour être rentable. Il propose à l'auteur de supprimer les solutions des exercices, ou encore de ne pas mettre d'images. Mais Knuth refuse, arguant qu'il a choisi cette maison d'édition en raison de la qualité de ses livres. Addison-Wesley propose alors de diviser l'ouvrage en sept volumes, afin de les vendre les uns après les autres à des prix raisonnables (*ibid.*)

2.2. LES VOLUMES

La première édition du premier volume, intitulé *Fundamental Algorithm* (KNUTH 1997a), est publiée en 1968. Ce premier volume se vend deux fois plus cher que les autres ouvrages d'informatiques de l'époque, mais

rencontre malgré tout un grand succès et est adopté comme *textbook*¹⁷ par cinquante universités dès l'année de sa parution (KNUTH 2020c). Il en est actuellement à son 42^e tirage, s'est vendu partout dans le monde, à plus de 400 000 exemplaires rien qu'en langue anglaise. Ce livre a établi le domaine de l'analyse d'algorithmes. Il réunit ainsi des contributions de divers auteurs sur les langages de programmation. Le but de cet ouvrage n'est pas seulement d'apprendre au lecteur comment utiliser le langage d'un autre, mais aussi d'apprendre à écrire un meilleur langage lui-même. On y retrouve donc beaucoup d'informations sur les systèmes de programmation, les logiciels existants et des analyses (D'AGOSTINO 2020).

La première édition du volume 2 (KNUTH 1997b) suit en 1969 et celle du volume 3 (KNUTH 1998) — consacré aux algorithmes de tris — en 1973. La première partie (A) du volume 4 — portant sur les algorithmes combinatoires — commencée en 1974, n'est parue qu'en 2011 (KNUTH 2011, 2020b). En effet, au cours des années 1970 et des décennies suivantes, il y a une croissance exponentielle de travaux concernant les algorithmes combinatoires, ce qui donne un coup de vieux aux écrits de Knuth au fur et à mesure qu'il les rédige. Aujourd'hui âgé de 83 ans (en 2021), Knuth travaille toujours activement sur la partie B du volume 4. Il distribue des sous-volumes, appelés fascicules, du volume 4. Par exemple le *Volume 4, Fascicule 5* couvrant entre autres le *backtracking* et les *dancing links* (KNUTH 2015, 2019; ROBERTS 2018).

2.3. L'IMPACT

Cette collection de livres représente l'œuvre d'une vie, et est en cours d'écriture depuis plus de cinquante ans. Knuth pense qu'il faudrait encore une vingtaine d'années pour qu'il puisse achever les sept volumes. Cette situation est le résultat de l'exigence d'exhaustivité de Knuth, ainsi que des profonds bouleversements et des très nombreuses avancées que l'informatique a connue au cours des dernières décennies. En 2018, lors d'une interview, Knuth en plaisante en disant : « Heureusement que je n'ai pas commencé en 1966 » car il y aurait alors eu encore plus d'évolutions à développer (FRIDMAN 2019).

Ces ouvrages sont une référence dans le domaine de l'informatique : quoique les premiers volumes soient aujourd'hui anciens, ils sont toujours

17. Livre servant de référence dans un cours universitaire américain.

vendus, ce qui est exceptionnel pour une science qui évolue si rapidement. Cela s'explique par l'approche fondamentale que l'auteur intègre à son travail : il s'agit en effet d'expliquer la théorie des algorithmes et de la programmation en général, en s'appuyant sur les principes des mathématiques discrètes (faisant intervenir uniquement des nombres entiers, qui s'opposent aux mathématiques du continu). D'autre part, Knuth est extrêmement pédagogue, et essaie d'expliquer chaque point de trois façons différentes afin que chacun puisse y trouver un élément qui lui parle et l'aide à comprendre (*ibid.*) Il donne de nombreux exemples concrets, et pense vraiment à son lecteur pendant le processus d'écriture. En effet, outre les exemples de code, on trouve dans le premier volume 850 exercices dont 90 % des réponses sont données (KNUTH 2020b). Ce ne sont pas pour autant des ouvrages simples à lire : après 652 pages, le premier tome se termine par un texte de Bill Gates sur la quatrième de couverture : « Vous devriez certainement m'envoyer un cv si vous pouvez tout lire » (ROBERTS 2018).

Non seulement ces ouvrages sont un manuel pratique qui permet aux étudiants d'apprendre « l'art de la programmation », au travers de nombreux exemples d'algorithmes — ces recettes qui nourrissent l'ère numérique — mais c'est aussi un livre d'histoire informatique. Les quatre volumes — qui sont aujourd'hui vendus sous forme de coffret pour environ 250 dollars — ont d'ailleurs été intégrés par *American Scientist* en 2013 dans sa liste des livres qui ont façonné le siècle dernier de la science, aux côtés d'une édition spéciale de *The Autobiography of Charles Darwin* ou encore des monographies d'Albert Einstein sur la relativité. Avec plus d'un million d'exemplaires imprimés, *L'Art de la programmation informatique* est donc une série d'ouvrages majeurs en sciences, aujourd'hui considérée comme une bible de son domaine (*ibid.*)

3. T_EX ET METAFONT

En 1977, la maison d'édition Addison-Wesley propose de rééditer le deuxième volume de *The Art of Computer Programming*. Knuth est enchanté par le projet : il complète même la première version avec les dernières innovations.

Cependant, il déchanté en voyant les épreuves de cette seconde édition : il les trouve horribles. Le noir n'est pas uniformément réparti, et les formules mathématiques ne sont pas belles à ses yeux. Il va alors trouver

l'éditeur, qui comprend son point de vue, mais lui dit qu'il ne peut pas faire grand-chose car la technique d'impression a évolué : alors que la première édition avait été imprimée avec un système Monotype, qui implique beaucoup de travail à la main, la seconde doit paraître à l'ère de l'impression par technique photographique. Cette situation ennuie terriblement Knuth, qui appréciait particulièrement son éditeur en raison du soin typographique et visuel apporté aux ouvrages. En effet, Addison-Wesley employait un typographe (Hans Wolf) qui avait appris son art en Allemagne et maîtrisait toutes les anciennes techniques typographiques, qu'il avait perfectionnées pour rendre très beaux les ouvrages de la maison d'édition (KNUTH 2020c).

Knuth cherche alors un moyen de rendre son livre plus agréable visuellement. Il voit un livre écrit par Patrick Winston sur l'intelligence artificielle qui a été imprimé numériquement en Californie. Cette technique d'impression n'a pas bonne réputation à l'époque — on dit qu'elle est de mauvaise qualité. Pourtant, Knuth trouve cet ouvrage bien plus lisible que les épreuves de sa seconde édition. D'autant plus que l'impression numérique lui est familière : il s'agit d'informatique, de donner des instructions sous forme de zéros et de uns à l'imprimante, pour lui dire si elle doit (1) ou non (0) déposer de l'encre sur tel ou tel pixel (*ibid.*)

Cela fait germer chez Knuth l'idée de créer son propre programme informatique pour imprimer numériquement ses ouvrages. Il pense ainsi être à l'abri de tout nouveau changement technologique qui pourrait frapper son éditeur. Il pense alors utiliser des technologies déjà existantes, notamment au centre de recherche Xerox de Palo Alto, où il sait que des caractères peuvent être formés. Cependant, après s'être renseigné, il découvre que Xerox lui permettrait d'utiliser son matériel de pointe dans le domaine, mais conserverait la possession de tous les caractères qu'il pourrait créer. Cela entre en opposition complète avec ses principes : il considère en effet que les lettres et les chiffres appartiennent à tous, et ne peuvent pas être possédés par une entreprise (*ibid.*)

3.1. L'AVENTURE TYPOGRAPHIQUE

Il décide donc de se lancer dans cette aventure typographique sans matériel de pointe. Il a cependant un avantage : il sait à quoi il veut que ses lettres ressemblent. En effet, son idée initiale est de se rapprocher autant que possible de la première édition de l'ouvrage, qu'il appréciait beaucoup.

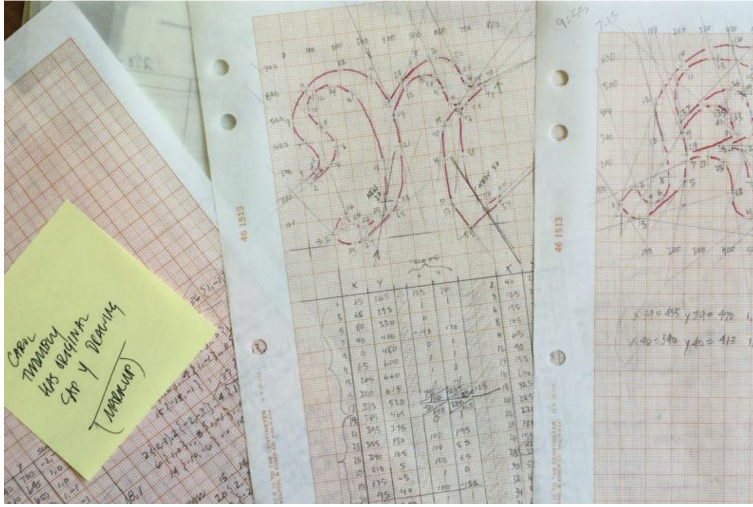


FIGURE 3. Modélisation de caractères.

Il fait donc des photographies des caractères de cette édition, et les agrandit en les projetant sur un mur avec un rétroprojecteur. Il essaie de reproduire les caractères qui lui apparaissent alors. Cependant, il est confronté à un problème : les tout petits caractères photographiés n'apparaissent pas nets à la projection. Il remarque alors qu'il existe des schémas répétitifs entre les différents caractères. Par exemple, un « n » occupe exactement deux fois l'espace d'un « i », alors qu'un « m » représente trois fois cette largeur. De même, un « u » a exactement la même largeur qu'un « n ». Il en déduit qu'il pourrait tracer des caractères en utilisant des formules mathématiques. En effet, comme nous l'avons vu précédemment, il a toujours été fasciné par le lien entre une équation et une courbe tracée graphiquement. Afin de définir avec précision les formules qu'il lui faudra utiliser, il décide d'approfondir ses connaissances en matière de typographie, en allant voir des spécimens conservés en Californie, tels qu'une *Bible* de Gutenberg préservée à Sacramento, et en assistant à des expositions sur l'histoire de la typographie (*ibid.*)

Il se met donc à rechercher des formules lui permettant de tracer chaque caractère. Il y parvient assez facilement pour 25 des lettres de l'alphabet,



FIGURE 4. Donald Knuth en dialogue avec le typographe Hermann Zapf.

mais il rencontre des difficultés avec la lettre « s », qui ne présente que des courbes, avec une inversion de courbure en son centre. Il finit par songer à composer le « s » à partir d'ellipses et de leurs tangentes. Il réussit donc à créer un alphabet constitué de courbes correspondant à des équations géométriques, ce qui lui permet de modifier des paramètres (tels que l'épaisseur du trait pour modifier la graisse, ou encore la taille du caractère) à partir d'une seule équation. Il réalise par la suite qu'il s'agit d'un raisonnement de pur informaticien ou mathématicien, mais que les typographes et les employés de l'industrie graphique n'utilisent jamais le terme de *paramètre*, et qu'ils pensent que Knuth souhaite qu'ils modifient le *périmètre* de ses caractères (KNUTH 2020c).

Knuth travaille à cette nouvelle police de caractères en étroite collaboration avec des typographes. Il demande l'aide de trois des quatre plus grands experts du domaine des années 1970–1980 pour corriger le *design* de ses nouveaux caractères, avant de tenter de passer à leur informatisation. Il s'agit de Chuck Bigelow, Matthew Carter et Hermann Zapf. Ceux-ci regardent ses dessins, et les annotent en rouge. Knuth modifie ses équations jusqu'à ce que les trois typographes soient satisfaits. Cette collaboration entre in-

formatique et typographie plaît beaucoup à Knuth, et elle se poursuit les années suivantes à travers une collaboration entre le département d'art et celui d'informatique de l'Université de Stanford. En 1984, par exemple, Richard Southall, Chuck Bigelow et Donald Knuth co-enseignent le *design* des polices de caractères au sein d'un cours commun (*ibid.*)

Afin d'implémenter ses caractères sur ordinateur, il rédige pour le dessin de chaque caractère un programme fondé sur ses caractéristiques mathématiques, écrit en langage ALGOL, très proche du langage mathématique usuel. Il cherche cependant à aller plus loin, et à harmoniser son travail en créant un système capable de prendre en compte tous les caractères et de modifier divers paramètres pour modifier la graisse, la taille, l'angle des courbures, etc. Pour créer ce système complet, il doit inventer un nouveau langage de description de formes géométriques dessinées. Il décide de l'appeler METAFONT, car il va au-delà (préfixe grec *meta*) d'une simple police de caractères (*font* en anglais). Il le publie en 1979 (*ibid.*)

3.2. DÉVELOPPEMENT DE T_EX

En parallèle de ce travail sur les polices de caractères se pose à lui un autre problème : il ne sait pas comment mettre en page son texte, car, en 1978, il n'existe que très peu de logiciels de traitements de texte, et ceux-ci sont destinés à un usage journalistique professionnel (ils sont par exemple utilisés par des journaux tels que *US News & World Report* et *Newsday*). Knuth décide donc de créer son propre logiciel lui permettant de mettre en page son deuxième volume comme il le souhaite. Il écrit sur papier les opérations à effectuer sous forme d'algorithmes, avec des spécifications précises pour tous les éléments qui lui semblent importants, tels que les formules mathématiques, les lignes de code, les tableaux, les figures et tous les caractères typographiques. Puis il confie à ses étudiants le soin de transcrire cela en un langage interprétable par la machine sous forme de zéros et de uns. Il est très surpris lorsqu'il constate un mois plus tard que les étudiants n'ont que très peu avancé. Il comprend alors que les spécifications qui lui semblaient si claires ne l'étaient pas pour les développeurs et qu'ils perdaient du temps. Il en déduit que le *designer* et le développeur du logiciel doivent être une seule et même personne : lui-même (*ibid.*)

Cependant, le travail de ses étudiants lui donne une idée de la structure que devra prendre son programme. Il décide de travailler avec un nouveau

type de langages, apparu dans les années 1970 : les langages structurés. Il trouve cela infiniment plus clair que tous les langages qu’il a utilisés précédemment, et ne ressent même pas le besoin de tester son programme pendant cinq mois tant il est certain qu’il est sur la bonne voie. Il ne commence à l’implémenter sur ordinateur et à le *debugger* qu’en mars 1978, et il peut voir le rendu de quelques pages aux alentours de Pâques. L’utilisation par lui-même et sa secrétaire, Phyllis Winkler (1933–2007), de ce logiciel lui permet de repérer de nouveaux problèmes, et de les régler aussi rapidement que possible. Il dit qu’il identifie une amélioration à apporter toutes les quatre pages tapées, et ce pendant 500 pages avant de parvenir à un résultat satisfaisant (KNUTH 2020c).

Il pense initialement n’écrire ce programme que pour ses propres ouvrages. Pourtant, les autres scientifiques travaillant dans le même laboratoire que lui voient les feuilles imprimées avec une grande qualité typographique, presque comme un vrai livre, et souhaitent utiliser son système. En particulier, le MIT se révèle intéressé, de même que l’*American Mathematical Society*, qui rencontre de sérieux problèmes typographiques pour ses revues (dont *Mathematical Reviews*) faisant intervenir nombre de formules mathématiques. Knuth, invité à intervenir lors de la prestigieuse conférence *Gibbs* en 1979 (à laquelle ne sont invités que d’éminents scientifiques) parle de son programme typographique, s’aperçoit que son audience est très attentive et qu’il existe un réel besoin d’un logiciel de traitement de texte permettant de mettre en forme les mathématiques (*ibid.*)

Knuth, qui n’avait pas anticipé ce succès, se retrouve donc à devoir écrire un manuel pour les utilisateurs de son logiciel. De plus, certains de ses collègues se mettent à travailler sur $\text{T}_{\text{E}}\text{X}$ et à l’améliorer. Cependant, Knuth se rend rapidement compte des limites d’une telle amélioration permanente : cela ne permet pas d’avoir une version stabilisée sur laquelle travailler de façon continue. Il décide donc de stopper le processus d’amélioration, et de sortir une version prototype appelée $\text{T}_{\text{E}}\text{X}$ 78. Une difficulté demeure : Knuth a écrit son programme dans un langage informatique utilisé uniquement à Stanford (nommé SAIL¹⁸) et ne pouvant être disponible que sur les ordinateurs PDP-10. Or d’autres universités, telles que le MIT, souhaitent se saisir du programme, mais ne peuvent pas l’employer en l’état. Des étudiants

18. *Stanford Artificial Intelligence Language*.

et informaticiens du MIT convertissent alors le code original de Knuth en langage Pascal, plus répandu et qui est reconnu par tous les ordinateurs de l'époque. Knuth propose même de payer une somme symbolique à ceux qui trouveraient une erreur dans le code, l'aidant ainsi à l'améliorer. De plus en plus d'utilisateurs et de développeurs se penchent sur ce programme. Il passe rapidement de 10 à 100 utilisateurs en juin 1978, puis à 1 000 en juillet de la même année, etc. L'engouement pour son logiciel croît exponentiellement (D'AGOSTINO 2020).

Au niveau du nom du logiciel, certains problèmes se sont posés : il existait déjà un système d'exploitation portant le nom de « Tex », produit par la compagnie Honeywell. Cette dernière refuse que le logiciel typographique porte le même nom. C'est pourquoi Knuth, qui avait déjà choisi l'appellation du programme, décide de la conserver, tout en modifiant la prononciation et en faisant croire qu'il s'agit d'un mot d'origine grecque et non anglo-saxonne : « T_EX » serait donc composé des trois caractères grecs « τ, ε, χ », et se prononce /tek/ et non /teks/ (KNUTH 2020c).

Après quatre années de retours d'utilisateurs, Knuth décide de réécrire tout le programme en prenant en compte les modifications à apporter. En particulier, il prend en compte les spécificités des langues européennes. En effet, le logiciel était prioritairement conçu pour les Américains, et ne prenait pas en compte certaines formes typographiques, telles que les accents (D'AGOSTINO 2020). De plus, il choisit pour cette réécriture un langage de programmation plus propice à la documentation, qui permette à un être humain de le lire aussi bien qu'à un ordinateur. En effet, il considère les langages de programmation comme toute autre forme d'écriture ; il compare d'ailleurs cela avec une partition de musique. Il assimile le code source de son programme à une élégante prose. Il espère d'ailleurs voir un jour un Prix Pulitzer être attribué au plus bel algorithme, au mieux écrit. Il pousse sa réflexion sur la littérature algorithmique jusqu'à publier l'intégralité du code source de cette dernière version de T_EX dans un livre de 600 pages, intitulé *T_EX: The Program* (KNUTH 1986a), qu'il pense relativement lisible par des êtres humains (KNUTH 2020c).

Toutes ces modifications l'amènent à concevoir la version T_EX 82, qui reste encore aujourd'hui très proche des versions les plus récentes de T_EX. Il s'agit de proposer au monde entier un logiciel stable, qui permette de retrouver la même mise en forme en 2002 qu'en 1982. Il n'est cependant pas opposé aux innovations et aux améliorations de son système : le code

source de $\text{T}_{\text{E}}\text{X}$ ayant été publié dans un livre, il considère que n'importe qui peut le réutiliser et le modifier, mais en utilisant un nom différent de $\text{T}_{\text{E}}\text{X}$. En effet, il souhaite que $\text{T}_{\text{E}}\text{X}$ demeure un point de repère stable là où les versions transformées de $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ par exemple sont amenées à évoluer (KNUTH 2020c). Son logiciel $\text{T}_{\text{E}}\text{X}$ marque le point de départ du mouvement des logiciels libres (puisque'il n'est pas propriétaire), mais différemment des licences actuelles telles que la *GNU Public License*, qui permet à tout un chacun de modifier le code source directement, sans changement de nom du programme. Il est néanmoins partisan de l'ouverture des codes sources, et fondamentalement opposé au brevet logiciel. Il a d'ailleurs adressé une lettre ouverte au bureau américain des brevets dans laquelle il explique que $\text{T}_{\text{E}}\text{X}$ n'aurait jamais pu voir le jour si ces brevets avaient été monnaie courante dans les années 1980 (FRAMALANG 2014).

3.3. AMÉLIORATIONS

Fier de toutes ses réalisations en termes typographiques et d'édition de texte, Knuth pense en 1981 avoir créé assez d'outils pour mettre harmonieusement en forme la deuxième édition de son deuxième volume. Il a obtenu des fonds de la part de deux organisations, la *National Science Foundation* et la *System Development Foundation*, qui lui ont attribué une dotation d'environ un million de dollars pour développer son projet. Grâce à ce budget confortable, il peut acquérir une imprimante AlphaType, qui a la meilleure résolution de l'époque. Après avoir imprimé quelques pages sur cette machine, il s'estime très satisfait du résultat, et pense que son livre sera parfait (KNUTH 2020c).

Pourtant, lorsque la maison d'édition Addison-Wesley lui envoie les épreuves, il est très déçu : si les lettres sont belles, les chiffres, eux, lui semblent horribles. Le chiffre « 2 », en particulier, est pour lui insupportable à regarder. En effet, Knuth a passé beaucoup de temps à *designer* et modifier ses lettres, mais s'est peu concentré sur les chiffres (*ibid.*)

Il réitère alors son travail de 1979 pour améliorer le résultat du programme METAFONT, mais cette fois en accordant son attention en priorité aux caractères numériques — en particulier les « 2 » et les « 5 », qui lui opposent les mêmes résistances que les « s » précédemment. En 1984, il finit par stabiliser la police de caractères *Computer Modern* — police par défaut du programme $\text{T}_{\text{E}}\text{X}$ aujourd'hui — qu'il estime alors « acceptables » (*ibid.*)

C'est ainsi qu'il a fallu à Knuth huit ans pour réaliser les programmes

T_EX et METAFONT, alors qu'il avait pensé travailler sur son problème typographique pendant un an. De ses travaux de cette période, il a tiré cinq ouvrages, constituant l'*opus* « *Computers & Typesetting* » :

- *The T_EX Book* (KNUTH 1984);
- *T_EX: The Program* (KNUTH 1986a);
- *The METAFONT Book* (KNUTH 1986d);
- *METAFONT: The Program* (KNUTH 1986b);
- *Computer Modern Typefaces* (KNUTH 1986c).

Grâce à ces travaux typographiques, Knuth a réussi à créer un logiciel permettant de mettre en forme des formules mathématiques sur mesure, parfaitement harmonieuses, comme il le souhaitait. Tous les travaux de Knuth à partir des années 1980 ont été rédigés sur T_EX. Nous avons choisi de reproduire quelques formules mathématiques de l'un de ses articles (KNUTH 1992) pour montrer toute la qualité du logiciel pour la production de telles suites de caractères.

$$F_n(x) = \frac{x(x-s)(x-2s)\dots(x-(n-1)s)}{n!}$$

$$\frac{(x+y)^n}{n!} = \sum_{k=0}^n \frac{x^k}{k!} \frac{y^{n-k}}{(n-k)!}$$

$$\frac{1}{n!} \sum_{k=0}^{n-1} \binom{n-1}{k} x^{\overline{k+1}} n^{n-1-k}$$

4. AUTRES TRAVAUX ET CENTRES D'INTÉRÊT

Donald Knuth est considéré comme un des maîtres des algorithmes. Son nom est ainsi associé à certains des spécimens les plus importants du domaine, tels que l'algorithme de recherche de chaînes Knuth-Morris-Pratt. Ce dernier, conçu en 1970, permet de trouver de façon optimale en temps toutes les occurrences d'un mot ou d'un motif de lettres donné dans un texte — par exemple, lorsque vous appuyez sur « contrôle + F » pour rechercher un mot-clé dans un document (ROBERTS 2018).

Mais Knuth a de multiples autres centres d'intérêt, à commencer par les nombres surréels.

En 1973, lors d'une année d'enseignement en Norvège, il décide d'écrire un roman sur les nombres surréels, concept de mathématiques fondamentales établi par John Conway qui regroupe les nombres réels (c'est-à-dire

les nombres entiers relatifs, ainsi que les nombres à virgule, fractionnaires, sans oublier des nombres tels que $\sqrt{2}$ ou π), les ordinaux transfinis¹⁹ et leurs inverses, respectivement plus grands et plus petits que n'importe quel nombre réel positif (des nombres tendant vers l'infini, mais pouvant être classés car tendant plus ou moins rapidement vers l'infini).

Il rédige ce roman en quelques jours (six jours pour un premier jet, puis une semaine de plus après des corrections de plusieurs relecteurs, notamment John Conway). Le livre, illustré par sa femme Jill, est publié en 1974 aux éditions Addison-Wesley Professional sous le titre *Surreal Numbers: How Two Ex-Students Turned on to Pure Mathematics and Found Total Happiness*²⁰ (KNUTH 1974). Long d'une centaine de pages, il est destiné à des étudiants qui souhaiteraient découvrir par eux-mêmes ces concepts mathématiques pointus. Cette histoire n'a pas fait l'unanimité dans la communauté scientifique, certains trouvant cette démarche géniale, et d'autres pensant cela ridicule. Quoiqu'il en soit, Knuth a beaucoup apprécié cette expérience, mais n'a jamais souhaité réitérer en écrivant d'autres romans mathématiques (KNUTH 2020c).

Il est tout autant passionné par les problèmes algorithmiques que par la musique (D'AGOSTINO 2020). Il aime en particulier jouer de l'orgue. Dans sa propre maison, il dispose d'un orgue dont il a lui-même dirigé la construction. Knuth nie cependant avoir tout talent particulier pour jouer de cet instrument. Il a composé une œuvre musicale multimédia pour orgue à tuyaux de 90 minutes, une traduction littérale du livre de l'Apocalypse de la *Bible* intitulée *Fantasia Apocalyptica* (KNUTH 2020a).

Croyant luthérien, très engagé dans la vie de son église, il a également

19. On montre en théorie des ensembles qu'il existe *autant* d'entiers naturels que de *couples* d'entiers naturels. Autrement dit, les deux ensembles \mathbb{N} (les entiers naturels) et $\mathbb{N} \times \mathbb{N}$ (les *couples* d'entiers naturels) ont le même *cardinal*. La définition des *ordinaux* permet des distinctions plus fines : par exemple, si ω est l'ordinal de \mathbb{N} — c'est le premier ordinal *transfini* —, nous pouvons l'interpréter comme un nombre « fictif » *après* l'infini des entiers et ranger les couples $(0, 0)$, $(0, 1)$, ... aux places n^{os} $0, 1, \dots$ puis les couples $(1, 0)$, $(1, 1)$, ... aux places n^{os} $\omega, \omega + 1, \dots$. Il ne s'agit que d'une généralisation de l'ordre lexicographique, utilisé dans un dictionnaire. Finalement, l'ordinal associé à l'ensemble $\mathbb{N} \times \mathbb{N}$ de couples est ω^2 .

20. *Les nombres surréels : comment deux ex-étudiants se mirent aux mathématiques pures et trouvèrent le bonheur total.*

publié *Things a Computer Scientist Rarely Talks About*²¹ (KNUTH 2001) sur les convictions religieuses et les sciences.

Responsable d'un groupe biblique dans sa paroisse luthérienne, il a expérimenté une lecture transversale de la *Bible* à la manière d'un statisticien. Fin connaisseur des écritures bibliques, il a publié, le livre *3:16 Bible Texts Illuminated* en 1990 (KNUTH 1990). Traduit en français en 2017, *3.16 la Bible en lumière* est une proposition de découvrir la *Bible* à travers chaque verset 16 des chapitres 3 des livres qui composent ce texte religieux. L'objectif de Knuth est ici de faire comprendre le livre en évitant de commenter son intégralité. Le principe premier est le hasard et le travail final est très cohérent. Chaque verset est accompagné d'une calligraphie produite par un groupe dirigé par Hermann Zapf. Spécialiste des études bibliques, Donald Knuth connaît parfaitement l'hébreu, l'araméen, le grec et le latin (BAYARD ÉDITIONS 2020).

Donald Knuth ménage son temps. Il prend sa retraite à 55 ans, limite ses engagements publics. La gestion du temps est la caractéristique déterminante de ce professeur depuis les années 1970. Généralement, sa disponibilité est l'heure de la « sieste », entre 13 h et 16 h. À l'époque où les ordinateurs étaient partagés entre les utilisateurs et fonctionnaient plus vite la nuit, il passait du jour à la nuit, décalant son emploi du temps, donnant rendez-vous à ses étudiants les vendredis de 20 h à minuit. De plus, Knuth a cessé d'utiliser le courrier électronique en prétendant qu'il s'en était servi entre 1975 et le 1^{er} janvier 1990, et que cela suffisait pour toute une vie. Il trouve plus efficace de tenir une correspondance en batch mode et y consacrer une journée tous les trois mois, en répondant par courrier « classique » (ROBERTS 2018).

5. CONCLUSION

Donald Knuth peut être considéré comme un véritable génie de l'informatique et de la programmation. Informaticien dès les débuts de la discipline, il est le père de l'analyse d'algorithmes et considère que « l'optimisation [informatique] est vraiment un art ». Passionné de mathématiques et d'anglais, il apprécie le langage sous toutes ses formes et introduit la notion de *literate programming* dans le domaine de l'informatique et de l'algorithmique. Il estime qu'« un bon codage est le synonyme d'une belle

21. *Ces choses dont un informaticien parle rarement.*

expression ». Perfectionniste notoire, il crée le logiciel $\text{T}_{\text{E}}\text{X}$, ainsi que sa police de caractères par défaut *Computer Modern*, et le langage permettant de générer et modifier des polices METAFONT, afin d'obtenir la mise en page et l'élégance typographique qu'il recherche. Autre exemple de son perfectionnisme, il a longtemps offert une prime de 2,56 dollars pour chaque erreur typographique découverte dans ses livres sous prétexte que « 256 cents font un dollar hexadécimal » (et, pour les erreurs de son ouvrage *3:16 Bible Texts Illuminated*, la prime passe à 3,16 dollars). Les informaticiens mentionnent cette récompense comme s'il s'agissait du prix Nobel d'informatique.

Après avoir vécu les débuts difficiles de l'informatique, qui avait du mal à s'imposer comme une discipline universitaire dans les années 1960, il exprime aujourd'hui sa perplexité sur la place prise par l'informatique dans la société. Il dit : « au début, les informaticiens craignaient que personne ne nous écoute. Maintenant, je crains que trop de gens nous écoutent ».

CRÉDITS PHOTOGRAPHIQUES

Figure 1 : Dasha Slobozhanina; figure 2 : Jill Knuth; figure 4 : Chuck Painter, Stanford News Service, remerciements à Pamela Moreland (Stanford News Library). Les photos des figures 1 à 3 proviennent du site *Web* de (KNUTH 2021); toutes les photos sont libres de droits.

RÉFÉRENCES

- BAYARD ÉDITIONS (2020). *Auteur Donald Knuth*. <https://www.bayard-editions.com/nos-auteurs/donald-knuth> (visité le 22/11/2020).
- D'Agostino, Susan (04/16/2020). "The Computer Scientist Who Can't Stop Telling Stories". In: *Quanta Magazine*. <https://www.quantamagazine.org/computer-scientist-donald-knuth-cant-stop-telling-stories-20200416/> (visited on 11/22/2020).
- FRAMALANG (11/08/2014). *Brevets logiciels : la position de Donald Knuth en 1995*. <https://framablog.org/2014/08/11/donald-knuth-brevets-logiciels-1995/> (visité le 22/11/2020).
- Fridman, Lex (2019). *Donald Knuth: Algorithms, Complexity, and The Art of Computer Programming*. Vidéo publiée sur YouTube. <https://www.youtube.com/watch?v=2BdBfsXbST8&t=2s>.

InformIT (2020). *Donald Ervin Knuth*.

<https://www.informit.com/authors/bio/3b944909-9332-403e-b7d1-5bd9c96e26fe> (visited on 11/22/2020).

Knuth, Donald Ervin (06/1957). “The Potrzebie System of Weights and Measures”. In: *Mad* 26.

- (1963). “Finite Semifields and Projective Planes”. PhD thesis. Pasadena, California: California Institute of Technology.
- (1974). *Surreal Numbers: How Two Ex-Students Turned on to Pure Mathematics and Found Total Happiness; A Mathematical Novelette*. Addison-Wesley Professional. ISBN: 0-201-03812-9.
- (1984). *The TeXbook*. Vol. A. Computers & Typesettings. Reading, Massachusetts: Addison-Wesley. ISBN: 0-201-13447-0.
- (1986a). *TeX: The Program*. Vol. B. Computers & Typesettings. Reading, Massachusetts: Addison-Wesley. ISBN: 0-201-13437-3.
- (1986b). *METAFONT: The Program*. Vol. D. Computers & Typesettings. Reading, Massachusetts: Addison-Wesley. ISBN: 0-201-13438-1.
- (1986c). *Computer Modern Typefaces*. Vol. E. Computers & Typesettings. Reading, Massachusetts: Addison-Wesley. ISBN: 0-201-13446-2.
- (1986d). *The METAFONTbook*. Vol. C. Computers & Typesettings. Reading, Massachusetts: Addison-Wesley. ISBN: 0-201-13445-4.
- (1990). *3:16 Bible Texts Illuminated*. Madison, Wisconsin: A-R Editions. ISBN: 0-89579-252-4.
- (1992). “Convolution Polynomials”. In: *Mathematica J.* 2.4, pp. 67–78. <https://arxiv.org/pdf/math/9207221v1.pdf>.
- (1997a). *Fundamental Algorithms*. 3rd ed. Vol. 1. The Art of Computer Programming. Reading, Massachusetts: Addison-Wesley. ISBN: 978-0-201-89683-1.
- (1997b). *Seminumerical Algorithms*. 3rd ed. Vol. 2. The Art of Computer Programming. Reading, Massachusetts: Addison-Wesley. ISBN: 978-0-201-89684-8.
- (1998). *Sorting and Searching*. 2nd ed. Vol. 3. The Art of Computer Programming. Reading, Massachusetts: Addison-Wesley. ISBN: 978-0-201-89685-5.
- (2001). *Things a Computer Scientist Rarely Talks About*. Vol. 136. CSLI²² Lecture Notes. ISBN: 1-57586-326-X.

22. *Center for the Study of Language and Information*.

- Knuth, Donald Ervin (2011). *Combinatorial Algorithms, Part 1*. 1st ed. Vol. 4A. The Art of Computer Programming. Reading, Massachusetts: Addison-Wesley. ISBN: 978-0-201-03804-0.
- (12/2015). *Satisfiability*. Vol. 4, Fascicle 6. The Art of Computer Programming. Reading, Massachusetts: Addison-Wesley. ISBN: 978-0-13-439760-3.
 - (11/2019). *Mathematical Preliminaries Redux; Backtracking; Dancing Link*. Vol. 4, Fascicle 5. The Art of Computer Programming. Reading, Massachusetts: Addison-Wesley. ISBN: 978-0-13-467179-6.
 - (2020a). *Fantasia Apocalyptica*. Interview.
<https://www-cs-faculty.stanford.edu/~knuth/fant.html>
 (visited on 11/22/2020).
 - (2020b). *Home Page*.
<https://www-cs-faculty.stanford.edu/~knuth/> (visited on 11/22/2020).
 - (2020c). *Web of Story*.
<https://www.webofstories.com/people/donald.d.knuth/1?o=SH>
 (visited on 11/22/2020).
 - (2021). *Downloadable Graphics*. <https://www-cs-faculty.stanford.edu/~knuth/graphics.html> (visited on 06/16/2021).
- O'Connor, J. J. and E. F. Robertson (2015). *Donald Knuth—Biography*.
<https://mathshistory.st-andrews.ac.uk/Biographies/Knuth/> (visited on 11/22/2020).
- Roberts, Siobhan (12/2018). “The Yoda of Silicon Valley”. In: *The New York Times*. ISSN: 0362-4331.
<https://www.nytimes.com/2018/12/17/science/donald-knuth-computers-algorithms-programming.html> (visited on 11/22/2020).

☛ Sandrine CHENEVEZ

Professeure-documentaliste dans l'enseignement secondaire depuis plusieurs années. Engagée dans la transmission des savoirs, elle réalise cette année un *Master 2* « Publication numérique » à l'ENSSIB avec l'objectif de s'impliquer dans la publication scientifique et l'écriture numérique.

☛ Stéphanie LOUISON

Après des études d'enseignement en physique-chimie, s'est ré-orientée vers la publication numérique, qu'elle étudie actuellement à l'ENSSIB de Lyon, pour travailler par la suite dans le domaine de l'édition scolaire.

☛ Myriam HAMLA

Titulaire d'un *Master* en sociologie, a choisi de compléter sa formation en choisissant la publication numérique à l'ENSSIB. Elle s'oriente désormais vers l'édition scientifique.