



LATEX

LE P'TIT COURS

Y. Tanguy

« Encre n. : vil composé de tanin, de noir de fumée, de gomme arabique et d'eau, principalement utilisé pour faciliter la contagion de la bêtise et promouvoir le crime intellectuel. »

Ambrose BIERCE, *The Devil's Dictionary* (1911)

Avant de commencer

Hélas pour vous, le rédacteur de ce qui suit avait envie de se manifester et de vous raconter « sa vie, son œuvre » dans un subtil texte introductif. Cependant, pour des raisons manifestes de budget papier, votre serviteur a dû se limiter à la seule genèse de ce document.

Lors de la première année de cette « conférence » sur \LaTeX , un élève avait en effet suggéré qu'un polycopié compléterait le cours à merveille. Ce qui a donné ce que vous avez sous les yeux. Toutefois, l'idée à la base de ce document ne consiste pas à faire un parfait polycopié de cours de \LaTeX . Oh que non ! À cela deux raisons :

- sur l'Internet et en librairie existent d'excellents documents sur le sujet ;
- de façon plus pragmatique, votre rédacteur honni doit sacrifier son temps sur l'autel du travail et du roupillon crapuleux qui s'ensuit... ce qui lui laisse, hélas, fort peu de temps pour faire le polycopié ultime en 14 volumes.

L'idée qui sous-tend ce document est toute autre : vous disposez ici d'un *exemple*, structuré en deux parties. La première partie revient à ce document final restituant nos choix de mise en forme. La seconde partie est tout simplement le code source du document. Ce code utilise la plupart des notions et commandes présentées en y ajoutant à quelques reprises de légères subtilités pour les plus curieux d'entre vous.

Par ailleurs, le peu d'exhaustivité de ce document est compensé à chaque fois que possible par des renvois à des documentations de référence (tel [minitoc](#)), des [liens hypertexte](#), de la bibliographie ou même des renvois à des pages de trois ouvrages de référence sous la forme suivante : Ces titres — respectivement de Bernard DESGRAUPES[13], Michel GOOSSENS[16] et Christian ROLLAND[31] — sont référencés en bibliographie, en compagnie d'autres ouvrages traitant de \LaTeX , de typographie, de ponctuation [10] et même de couleur [28] !

Bien évidemment, ce qui suit doit beaucoup aux nombreuses informations, aides et échanges trouvés sur l'Internet. Un grand merci aux passionnés de \LaTeX cachés derrière tout cela !

»Réfs
[13]32–34,487–501
[16]24–47
[31]40–41

Sommaire

Avant de commencer	ii
Sommaire	iii
Liste des tableaux	viii
Liste des figures	ix
Liste d'exemples de code	x
I Notions fondamentales	1
1 Une histoire de fous	2
1.1 Au commencement était T _E X	2
1.1.1 Le contexte	2
1.1.2 Le résultat	3
1.2 Puis vint L ^A T _E X	3
1.3 Longtemps après vinrent X _Y T _E X et X _Y L ^A T _E X	4
1.4 Et LuaT _E X et LuaL ^A T _E X les rejoignirent	4
1.5 Un souci	4
2 Le fonctionnement de T_EX	5
2.1 Un fichier L ^A T _E X	5
2.1.1 La structure	5
2.1.2 Un exemple	6
2.2 La « grammaire » de L ^A T _E X	6
2.2.1 Les caractères actifs	6
2.2.2 Les groupes	7
2.2.3 Les différentes commandes	7
2.2.4 Unités de mesure	8
2.3 La compilation	8
2.3.1 La mécanique de la compilation	8
2.3.2 Les fichiers générés par la compilation	9
2.4 La gestion des erreurs	9
2.5 L'impression	9
2.6 Des compilations alternatives	10
II Notions courantes	11
3 Préambule et classe du document	12
3.1 Un préambule minimal	12
3.1.1 Cas avec L ^A T _E X et pdfL ^A T _E X	12

3.1.2	Cas avec X _Y LaTeX et LuaLaTeX	12
3.2	La classe du document	13
3.2.1	Les grands classiques	13
3.2.2	Pour aller plus loin	13
3.3	Les autres paquets usuels	14
4	Le texte	15
4.1	Les caractères	15
4.1.1	La police de caractère	15
4.1.2	Le style des caractères	16
4.1.3	Les caractères spéciaux	17
4.1.4	Pour aller plus loin	19
4.2	Autour des caractères	19
4.2.1	Les espaces	19
4.2.2	La césure	20
4.2.3	Le soulignement	21
4.2.4	L'encadrement	22
4.3	La couleur	23
4.3.1	La définition des couleurs	23
4.3.2	L'utilisation de couleurs prédéfinies	23
4.3.3	L'appel des couleurs	25
4.4	L'agencement du texte	25
4.4.1	L'alignement	25
4.4.2	Les listes	26
4.4.3	Les notes	27
4.5	Les hyperliens	28
4.6	Pour aller plus loin	29
5	Les structures du document	30
5.1	Les divisions	31
5.1.1	Les divisions non numérotées	31
5.2	La table des matières	31
5.2.1	La profondeur de la table des matières	31
5.2.2	L'ajout en table des matières	32
5.2.3	Les mini-tables des matières	32
5.2.4	Pour aller plus loin	32
5.3	Les références	33
5.3.1	Les commandes usuelles	33
5.3.2	Pour aller plus loin	33
5.4	L'index	33
5.4.1	Le programme externe makeindex	33
5.4.2	Quelques compléments sur l'indexation	34
5.4.3	Pour aller plus loin	34
5.5	La bibliographie	34
5.5.1	Le fichier bibliographique	35
5.5.2	Le code LaTeX	35
5.5.3	Le contenu final	35
5.5.4	BibTeX	36
5.5.5	Pour aller plus loin	36
5.6	Les en-têtes et pieds de page	36
5.6.1	Les commandes usuelles	36
5.6.2	Pour aller plus loin	36
5.7	Le titre	36
5.7.1	Les commandes usuelles	36
5.7.2	Pour aller plus loin	37

6	Les mathématiques	38
6.1	Les modes mathématiques	38
6.1.1	Principes	38
6.1.2	Différents environnements pour équation	38
6.1.3	Numérotation des équations	39
6.2	Les symboles mathématiques	40
6.2.1	Opérateurs	40
6.2.2	Lettres	41
6.2.3	Flèches	41
6.2.4	Accents	42
6.2.5	Notations graphiques	42
6.3	Les exposants et indices	42
6.4	Les délimiteurs	43
6.5	Racines et fractions	44
6.6	Les textes et espaces	44
6.7	Les matrices	44
6.8	Un élément d'actuariat	46
7	Les flottants : tableaux et figures	47
7.1	Les tableaux	47
7.1.1	Les règles de base	47
7.1.2	Le passage en flottant	48
7.1.3	Les filets	49
7.1.4	La fusion de cellules	50
7.1.5	L'enrichissement du format-colonne	50
7.1.6	Un peu de couleur	51
7.1.7	Une saisie simplifiée	52
7.1.8	Pour aller plus loin	53
7.2	Les images	53
7.2.1	Le format et la localisation des images	53
7.2.2	L'insertion des images	53
7.2.3	Le passage en flottant	54
7.2.4	Commandes complémentaires	55
7.2.5	Pour aller plus loin	56
III	Notions avancées	57
8	La programmation avec \LaTeX	58
8.1	La scission du code	58
8.2	La définition de nouvelles commandes	58
8.2.1	Principes	58
8.2.2	L'espacement	60
8.2.3	Les mathématiques	60
8.2.4	Les compteurs	61
8.2.5	Les dimensions	61
8.2.6	Les boucles	62
8.2.7	Les conditions	62
8.2.8	Pour aller plus loin	63
8.3	La modification des réglages de \LaTeX	63
8.3.1	Les libellés de division	63
8.3.2	Les dimensions de la page	64
8.3.3	La personnalisation des en-têtes et pieds de page	66
8.3.4	La personnalisation de l'index	67
8.3.5	La gestion forcée des flottants	68

8.3.6	Pour aller plus loin	69
9	Les graphiques	70
9.1	PsTricks	70
9.1.1	Le placement	70
9.1.2	Les boîtes	71
9.1.3	Les grilles	71
9.1.4	Les points et courbes	72
9.1.5	Les flèches et options	72
9.1.6	Vers les mathématiques	73
9.1.7	Le tracé de fonction mathématique	73
9.1.8	Exemples capillotractés	75
9.2	Tikz	77
9.2.1	Le placement par les nœuds	78
9.2.2	Les boîtes ou formes	78
9.2.3	Les grilles	79
9.2.4	Les points et courbes	79
10	Les polices de caractères avec X_YLaTeX	81
10.1	Modifications associées à X _Y LaTeX	81
10.2	Pour aller plus loin	82
11	Un fichier d'extension	83
11.1	Déclarations	83
11.2	Options	83
11.2.1	Déclaration des options	83
11.2.2	Traitement des options	83
11.2.3	Transmission d'options	84
11.3	Appel d'autres paquets	84
11.4	Codes particuliers	84
11.5	Exemple	85
11.6	Pour aller plus loin	85
12	PythonTeX	86
12.1	Brève présentation du langage	86
12.2	Calcul scientifique	86
12.3	Exemples d'utilisation	86
12.4	Pour aller plus loin	86
IV	Annexes	87
A	Installation	88
A.1	Une installation compacte : USBTeX	88
A.1.1	Installation	88
A.1.2	Maintenance	88
A.1.3	La vérification orthographique	89
A.2	Une installation complète	89
A.2.1	LaTeX	89
A.2.2	Ghostsript et GSview	90
A.2.3	Configuration de Windows	91
A.3	Des éditeurs	92
A.3.1	Texmaker	93
A.3.2	TeXnicCenter	93

B	Liens avec d'autres programmes	95
B.1	Conversion d'image	95
B.2	Excel	96
B.3	Les logiciels mathématiques	97
B.3.1	R, avec Sweave	97
C	Un CV sous L^AT_EX	99
C.1	Une nouvelle classe	99
C.2	Différents thèmes	99
C.3	Paramètres personnels	99
C.4	Découpage du CV	100
C.5	Commandes complémentaires	100
C.6	Exemple	101
C.7	Pour aller plus loin	101
D	La page de garde	102
D.1	Définition et exemple	102
D.2	Appel de l'extension	104
D.2.1	L'option de filière	104
D.2.2	L'option de brouillon	104
D.2.3	Les options de confidentialité	104
D.2.4	L'option de page verso uniquement	104
D.2.5	L'option de liens hypertextes	104
D.3	Paramètres	105
D.4	Affichage	105
D.5	Commandes complémentaires	105
D.5.1	Commandes de texte	105
D.5.2	Commandes de sigle	105
D.5.3	Commandes avec lien hypertexte	106
E	Avant de réduire votre ordinateur en poussière	107
E.1	Quelques réflexes de survie	107
E.1.1	Réflexes pour amateurs	107
E.1.2	Réflexes pour passionnés	107
E.2	Le bestiaire monstrueux	107
F	Index des paquets et programmes	109
G	Index des concepts	111
	Bibliographie	112

Liste des tableaux

2.1	Caractères actifs	6
2.2	Unités de longueur absolues	8
2.3	Unités de longueur relatives	8
4.1	Différents styles de caractère	16
4.2	Différentes tailles de caractères	17
4.3	Caractères particuliers	17
4.4	Caractères particuliers issus de <code>babel</code>	18
4.5	Caractères particuliers issus de <code>textcomp</code>	18
4.6	Table de caractère associée à <code>pifont</code>	18
4.7	Espacements horizontaux	19
4.8	Espacements horizontaux	20
4.9	Différents soulignements	22
4.10	Systèmes de couleur courants	23
4.11	Différents styles de présentation	26
4.12	Différents styles de liste	26
5.1	Divisions d'un document	31
6.1	Différents styles de présentation d'équation	39
6.2	Opérateurs	40
6.3	Égalités et inégalités	40
6.4	Notations ensemblistes et fonctionnelles	40
6.5	Lettres grecques	41
6.6	Flèches	41
6.7	Flèches	42
6.8	Notations graphiques	42
6.9	Grands opérateurs	42
6.10	Délimiteurs	43
6.11	Espacements mathématiques	44
6.12	Différents styles de présentation de matrice	45
8.1	Quelques noms de variables	63
A.1	Étapes d'installation de la <code>TeXlive 2014</code>	90
A.2	Étapes d'installation de <code>Ghostscript</code>	91
A.3	Étapes d'installation de <code>GSview</code>	91
D.1	Les paramètres de <code>pagedegarde</code>	105
D.2	Commandes complémentaires de texte de <code>pagedegarde</code>	105
D.3	Commandes complémentaires de sigle de <code>pagedegarde</code>	106

Liste des figures

2.1	Du fichier <code>.tex</code> au fichier <code>.pdf</code>	10
4.1	Couleurs de base	24
4.2	Couleurs issues de l'option <code>dvipsnames</code>	24
8.1	Une page et ses dimensions	65
9.1	Quelques OAT	73
9.2	Pyramide des effectifs	75
9.3	Exemples de fonctions paramétriques	77
A.1	Page d'accueil de T _E XLive 2014	90
A.2	Exemple d'entrée des chemins sous Windows XP	92
A.3	Paramétrage de Texmaker	93
A.4	Paramétrage de TeXnicCenter	94
B.1	Formulaire d'exportation de The Gimp	95
B.2	Formulaire des options <code>.eps</code> de The Gimp	96
D.1	Sigles	106

Liste d'exemples de code

2.1	Exemple simple	6
2.2	Exemple de groupe	7
3.1	Un code minimal pour \LaTeX	12
3.2	Un code minimal pour \XeTeX et \LuaTeX	13
4.1	Tailles de caractères	17
4.2	Commandes issues de \pifont	19
4.3	Espaces horizontales	20
4.4	Césure dans le texte	21
4.5	Césure en préambule	21
4.6	Appel pour le soulignement	21
4.7	Une boîte encadrée	22
4.8	Une boîte de texte encadrée	22
4.9	Quelques encadrements	22
4.10	Définition de couleur	23
4.11	Quelques couleurs	23
4.12	Texte en couleur	25
4.13	Bascule de couleur	25
4.14	Boîtes en couleur	25
4.15	Page en couleur	25
4.16	Une liste simple	26
4.17	Une liste revisitée	27
4.18	Listes issues de \pifont	27
4.19	Appel du paquet \hyperref	28
4.20	Exemple de paramétrages	28
5.1	Modification de la profondeur de la table des matières	32
5.2	Ajout d'une entrée à la table des matières	32
5.3	Appel de \minitoc en français	32
5.4	Disposition des commandes d'index	33
5.5	Appel générique de \makeindex	34
5.6	Définition minimale d'un livre pour \BibTeX	35
5.7	Cas d'auteurs multiples pour \BibTeX	35
5.8	Commandes pour la bibliographie	35
6.1	Modes mathématiques	38
6.2	Equations et alignements	39
6.3	Mises en indice et en exposant	43
6.4	Délimiteurs redimensionnés	43
6.5	Racines et fractions	44
6.6	Exemple de matrice	45
6.7	Utilisation d'un tableau	45
6.8	Formules de mathématiques des assurances	46
7.1	Tableau simple	48
7.2	Tableau flottant	49
7.3	Tableau avec filets	49

7.4	Tableau avec fusion de cellules	50
7.5	Tableau avec égalisation des largeurs de colonne	51
7.6	Tableau avec cellules colorées	51
7.7	Tableau avec cellules et filets colorées	52
7.8	Tableau issu d'une commande Excel dédiée	53
7.9	Insertion d'images	53
7.10	Insertion d'une image dans du texte	54
7.11	Insertion d'une image	54
7.12	Insertion d'une image avec rotation	55
7.13	Boîtes avec <code>graphicx</code>	55
7.14	Exemple de déformations de texte	56
7.15	Commande d'insertion d'une image	56
8.1	Nouvelle commande	59
8.2	Définitions de commande sans argument	59
8.3	Définition de commande à un argument	59
8.4	Définition de commande à plusieurs arguments	60
8.5	Exemples d'utilisation de <code>\xspace</code>	60
8.6	Définition de commande mathématique protégée	61
8.7	Gestion de compteurs	61
8.8	Exemple d'utilisation d'un compteur	61
8.9	Gestion de dimensions	62
8.10	Boucle	62
8.11	Exemples de boucle	62
8.12	Condition	62
8.13	Exemple de test conditionnel	63
8.14	Redéfinition du titre de la table des matières	63
8.15	Modification de page par commande	66
8.16	Modification de page pour ce document	66
8.17	Appels pour l'application du style <code>fancy</code>	66
8.18	Redéfinition des textes d'en-tête pour chapitre et section	67
8.19	Contenu et présentation de l'en-tête	67
8.20	Gestion des filets et de l'espace de l'en-tête	67
8.21	Définition d'un en-tête vide	67
8.22	Redéfinition de l'index	68
8.23	Lettrines simples pour un index	68
8.24	Positionnement forcé d'une figure	69
9.1	L'environnement de <code>pstricks</code>	70
9.2	Placements et redimensionnements	71
9.3	Boîtes	71
9.4	Grilles	72
9.5	Points et courbes	72
9.6	Options sur points et courbes	73
9.7	Une sinusoïde	74
9.8	Une rosace	75
9.9	Construction d'une famille de courbes	76
9.10	Appel de la commande pour les courbes	76
9.11	L'environnement de <code>tikz</code>	78
9.12	Placements et redimensionnements	78
9.13	Boîtes	79
9.14	Grilles	79
9.15	Points et courbes	80
10.1	Création d'une famille de polices de caractères	81
10.2	Mise en valeur par défaut d'une famille de polices de caractères	82
11.1	Déclaration d'un paquet	83

11.2	Déclaration et définition d'une option	83
11.3	Exécution d'options listées	84
11.4	Exécution des options appelées par l'utilisateur	84
11.5	Transmission d'options	84
11.6	Appel de paquet	84
11.7	Stockage de code pour exécution à des moments spécifiques	84
11.8	Exemple	85
A.1	Code pour l'UTF8	93
B.1	Macro VB créant un code \LaTeX en encodage latin1	96
B.2	Macro VB créant un code \LaTeX en encodage UTF8	97
B.3	Encadrement d'un code R sous \LaTeX	97
B.4	Exemples de code R sous \LaTeX	98
B.5	Instructions sous R	98
B.6	Paquet nécessaire pour travailler avec R et Sweave	98
C.1	Définition de couleur pour le CV	99
C.2	Exemple de CV	101
D.1	Exemple de génération de la page de garde	102
E.1	Augmentation de capacité pour \LaTeX	108

Première partie

Notions fondamentales

Chapitre 1

Une histoire de fous

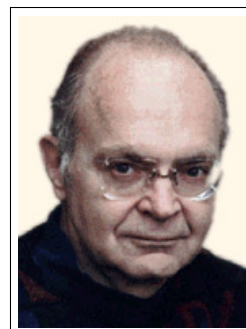
1.1 Au commencement était T_EX

T_EX (du grec $\tau\epsilon\chi$ qui a donné le mot « technique » et qui explique la prononciation « tèque ») est un programme de mise en forme de textes techniques, du simple article aux ouvrages en plusieurs tomes. Pour mieux comprendre l'aspect quasi-légeraire de T_EX, un peu d'histoire.

1.1.1 Le contexte

Durant les années 70, Donald E. KNUTH[19], un mathématicien américain, s'était attelé à l'écriture de ce qui est désormais une référence fondamentale (rien de moins) en informatique : *The Art of Computer Programming*. Dès l'origine, il avait planifié la sortie de plusieurs volumes sur une quarantaine d'années ! Or, en 1977, Donald KNUTH rencontra un problème très particulier¹.

« Pourquoi ai-je commencé à écrire T_EX en 1977 ? Cette histoire avait débuté longtemps auparavant, à l'occasion de la publication de mes livres The Art of Computer Programming. J'avais préparé une seconde édition du tome 2, mais lorsque je reçus les épreuves, ce fut horrible — la technique d'impression avait radicalement changé depuis la première édition. Les livres étaient maintenant composés à l'aide de photocomposeuses, au lieu des monotypes à plomb fondu ; en outre (hélas !), ces photocomposeuses étaient pilotées par des ordinateurs au lieu d'être supervisées « manuellement ». Il en résultait une gestion désastreuse des blancs, surtout quand il s'agissait de mathématiques, et les fontes étaient décevantes comparées aux anciennes.



J'étais désespéré et ne savais que faire. Addison-Wesley² m'offrit de tout recomposer à l'aide des vieilles monotypes, mais je savais que la vieille méthode de composition était en train de mourir rapidement ; j'étais certain que lorsque j'aurai fini le tome 4, le même phénomène se reproduirait de nouveau et je ne voulais pas d'un résultat qui ressemblerait aux épreuves que j'avais vues. (...)

Dès son début, en 1977, le projet de recherche T_EX dans lequel j'étais embarqué comportait deux axes principaux. Le premier était la qualité : nous ne voulions pas produire de bons documents, nous voulions qu'ils soient les meilleurs. (...)

Le second but recherché était l'archivage : il s'agissait de créer un système qui serait indépendant, autant que faire se peut, des mutations technologiques. Lorsqu'une nouvelle génération de machine à imprimer arriverait, je voulais être capable de maintenir le même

1. Ce texte, maladroitement traduit par votre serveur, a pour une source une conférence de Donald KNUTH en 1986, reprise *in extenso* dans le **septième numéro** de la revue TUGboat, page 95–98. À l'adresse <<http://www.webofstories.com/play/17110?o=MS>> se trouve une interview vidéo de Donald KNUTH sur ce thème.

2. Éditeur spécialisé dans les domaines scientifiques.

niveau de qualité au lieu de repartir à zéro. Je voulais produire quelque chose qui pourrait encore servir dans un siècle. En d'autres termes, mon but était de m'y prendre de telle manière que, si l'on sauvegardait les spécifications d'un livre, nos descendants seraient encore capables d'éditer le même livre en l'an 2086. »

1.1.2 Le résultat

La première version de \TeX date de 1978. Suite à corrections d'erreurs et à quelques évolutions, Donald KNUTH décida de figer \TeX en 1982 à la version 3.14159³.

En terme de qualité et conformément à l'idée initiale, \TeX a de quoi surprendre. Ainsi, son unité de longueur de référence la plus petite, le *scaled point*, équivaut à $5,4 \times 10^{-9}\text{m}$. Autrement dit, les caractères sont ici placés avec une précision dépassant largement la précision de l'œil humain puisque nous nous plaçons ici à une échelle de l'ordre de grandeur du spectre de la lumière visible. \TeX sera donc dépassé quand l'évolution nous aura fait nous passer de nos yeux !

En terme informatique, \TeX est à la fois un programme et un langage. \TeX est ainsi un programme convertissant un fichier texte en un document décrivant la manière de positionner et de présenter ce texte sur une suite de pages. Par ailleurs, \TeX est un langage composé de plus de 300 fonctions dites « primitives ». Ces fonctions, placées dans le fichier texte, permettent d'indiquer au programme \TeX comment présenter le texte.

Ces fonctions, généralement simples, ont le désavantage de ne pas permettre de traiter un texte aisément. Seule l'action consécutive de plusieurs d'entre elles permet d'obtenir des résultats complexes, mais naturels. Ainsi en est-il, par exemple, de la création d'une page titre : dans ce cas, le texte sera centré selon plusieurs critères (le format de la page en particulier), sera écrit plus grand, sera sur une feuille séparée des autres, feuille qui ne sera pas numérotée... De fait, \TeX permet de créer une nouvelle fonction de présentation par composition de fonctions primitives : une **macrocommande** ou macros, nommées dans toute la suite commandes. Bien entendu, de nouvelles commandes peuvent également s'obtenir par composition de commandes existantes.

Un ensemble de commandes pensées pour permettre de présenter un document forme un sur-ensemble de \TeX , un **format**. Le premier format créé par KNUTH, un format minimal, s'appelle *plain*.

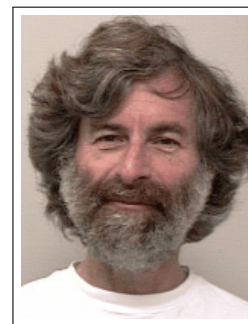
1.2 Puis vint \LaTeX

\LaTeX (prononcé « latèque »), écrit en 1982 par Leslie LAMPORT [21], est un format simplifiant l'usage de \TeX en le dotant de commandes exécutant des tâches importantes de présentation.

Par la suite, des ensembles de commandes ont été développés, ceci afin de développer tel ou tel aspect particulier de présentation : par exemple offrir des mises en page alternatives des titres. Un tel ensemble de commandes regroupées en un fichier est ici appelé une **extension** ou un « **paquet** » (en anglais *package*).

Et les besoins d'extension étaient vastes ! Les possibilités de \TeX par le biais de \LaTeX et de ses extensions ont, de fait, explosé. D'un logiciel pensé pour écrire des articles scientifiques assez stéréotypés, \TeX est devenu un logiciel pensé pour rédiger des documents très variés — de la simple lettre au livre en passant par les transparents ou des partitions — dans des domaines aussi différents que la physique, la chimie, les mathématiques, la linguistique, la musique, le jeu d'échecs, etc.

Depuis 1994, la version normalisée s'appelle $\text{\LaTeX 2}_{\epsilon}$ (qui se lit « latèque deux oeufs »). Elle est compatible (dans la mesure du possible) avec les anciens standards.



3. Chaque nouvelle version de \TeX ajoute des décimales de π au numéro de version.

1.3 Longtemps après vinrent X_ƎT_ƎX et X_ƎL^AT_ƎX

T_ƎX et L^AT_ƎX sont rapidement devenus des outils internationaux mais ils avaient quelques limitations, en particulier sur la gestion de langues étrangères et de polices de caractères. Ceci s'expliquait en particulier par une problématique d'encodage que Jonathan K_ƎW, avec X_ƎT_ƎX, fit grandement avancer.

Fondamentalement, les ordinateurs ne traitent que des nombres binaires. Un fichier texte ne contient pour ainsi dire qu'une chaîne de zéros et de uns. Pour que l'ordinateur nous restitue des caractères, il faut avoir recours à un **encodage** autrement dit une table de correspondance qui permet de dire que telle chaîne binaire est associée à tel caractère unique. Par exemple, en encodage ASCII, « 01000001 » correspond à la lettre « A ».

Cependant, il existe plusieurs encodages et ils ne sont pas forcément cohérents entre eux. Depuis 1991, un projet d'encodage mondial suit cependant son cours : Unicode⁴. Il a pour but de traiter les différentes langues de l'humanité en un unique encodage : s'y retrouvent des langues aussi variées que le japonais, le tamul, le tagalog, le khmer.

Du fait de ses travaux l'amenant à travailler avec des langues asiatiques, Jonathan K_ƎW⁵ proposa en 2004 une version de T_ƎX, nommée X_ƎT_ƎX (prononcé « zétex »), permettant de gérer l'Unicode et donc le multilinguisme.

X_ƎT_ƎX et **fontspec** — un paquet dédié à la gestion des polices de caractères — offrent en plus des fonctionnalités plus étendues (voir page 81) :

- accès généralisé aux polices OpenType Fonts (OTF) et TrueType Fonts (TTF) ;
- accès à des caractères alternatifs et à des ligatures spécifiques de ces polices ;
- gestion de la transparence des caractères ;
- simplification de la chaîne de compilation pour **pstricks**.



1.4 Et LuaT_ƎX et LuaL^AT_ƎX les rejoignirent

Développé par Taco HOEKWATER, Hartmut HENKEL et Hans HAGEN, LuaT_ƎX⁶ étend les possibilités du monde T_ƎX dans une nouvelle direction : intégrer un langage de script, Lua, dans le code T_ƎX. Ceci permet de faire exécuter à L^AT_ƎX par le biais de Lua des tâches plus complexes dans le traitement du texte : Lua peut en effet modifier le comportement de T_ƎX ou le compléter. De façon similaire, ceci permet de faire des calculs complexes avec T_ƎX, ce dernier étant relativement limité en ce domaine.

1.5 Un souci

L'actuelle variété du monde de T_ƎX a tout de même un prix dans la mesure où ces extensions et différentes versions de T_ƎX ont introduit des incompatibilités : différents paquets utilisent des noms de commandes identiques pour réaliser des tâches bien différentes. Du coup, une certaine prudence reste parfois de mise si de multiples paquets sont utilisés, ceci pour éviter des résultats erronés.

Cette situation a conduit au projet de normalisation L^AT_ƎX3, sous la direction des « gurus » L^AT_ƎX parmi lesquels se trouve Leslie LAMPORT⁷. L'histoire n'a donc pas fini de se poursuivre !

Vous cernez mieux la bête ? Passons alors à la T_ƎXnique !

4. Le site officiel d'Unicode présente très largement cet encodage : <<http://www.unicode.org/>>.

5. Une interview de Jonathan K_ƎW sur ce sujet : <<http://tug.org/interviews/kew.html>>.

6. Le site officiel est à l'adresse <<http://luatex.org/>>.

7. Pour plus de détails, consulter <<http://www.latex-project.org>>.

Chapitre 2

Le fonctionnement de T_EX

Décrire le fonctionnement précis et exact de T_EX, le programme derrière le format L^AT_EX, demanderait un ouvrage complet. En l'occurrence, cet ouvrage existe et s'appelle le T_EXbook[19] : il a été rédigé par K_NU_TH lui-même... un expert que nous ne chercherons pas à dépasser sur son sujet ! Nous nous limitons donc ici à une description partielle et partielle de la structure d'un document L^AT_EX, de sa composition et des conséquences de ces règles sur la façon de rédiger du code L^AT_EX¹.

2.1 Un fichier L^AT_EX

Un fichier L^AT_EX est un document lisible par n'importe quel éditeur de texte classique. Le texte et les commandes — toute chaîne de caractères commençant par un « \ » — qui y sont incluses restent lisibles, ceci en opposition avec de nombreux autres logiciels qui codent ces éléments.

Un fichier L^AT_EX a pour extension `.tex`² et son nom ne doit pas contenir d'espace ou de lettre accentuée. Il est traditionnellement dénommé fichier source ou **source**³, pour le distinguer du fichier présentant le résultat final.

2.1.1 La structure

Ce fichier est toujours constitué de deux parties :

- le **préambule**, situé en début de document. Il contient un ensemble de définitions génériques servant à présenter l'ensemble du document. Plus précisément, s'y trouvent réunis la définition du type de document, l'appel à des paquets spécifiques — fichiers définissant des commandes étendant les fonctionnalités de L^AT_EX — mais aussi la (re)définition de commandes spécifiques au document. Placer du texte libre dans cette partie provoque une erreur de L^AT_EX ;
- le **corps**. Il contient le texte qui sera affiché ainsi que des commandes modifiant la présentation de ce texte à l'endroit où elles sont placées ou sur la zone qu'elles encadrent.

Les deux parties sont séparées par la commande⁴ : `\begin{document}`. Le corps s'achève avec la commande `\end{document}`. Tout le texte placé au-delà de cette seconde instruction est purement et simplement ignoré par L^AT_EX.

1. Pour simplifier, T_EX et L^AT_EX seront systématiquement confondus par la suite.

2. Dans toute la suite, un « fichier d'extension tex » sera désigné comme « fichier `.tex` ».

3. Source est de plus en plus considéré comme un nom masculin dans ce cas précis.

4. L'usage de la fonte « machine à écrire » est quasiment systématique dans les ouvrages d'informatique lorsque est évoqué un élément du langage de programmation, élément qui serait tapé tel quel à l'écran. Est ici ajouté de la couleur pour le plaisir de complexifier les définitions des commandes permettant cet affichage.

2.1.2 Un exemple

Voici un exemple simple d'un fichier L^AT_EX :

```
\documentclass[a4paper]{article}
\usepackage{eurosym}
\begin{document}
Savais-tu que le signe de l'\emph{euro} se tape \euro?
\end{document}
```

CODE 2.1 – Exemple simple

Ici, le préambule se limite à deux lignes. La première ligne définit le type du document : elle demande à L^AT_EX de faire appel à des fichiers contenant des commandes et paramètres permettant de présenter le document. Ce point est détaillé au chapitre 3.

La seconde ligne charge un paquet⁵, **eurosym**, qui définit la commande `\euro`.

Dans le corps, constitué des trois lignes suivantes, se trouve le texte à afficher. Ce texte est localement modifié par la commande `\emph` qui fait passer le texte qu'elle contient entre accolades en italique. De plus, la commande `\euro` va permettre l'affichage du symbole €.

2.2 La « grammaire » de L^AT_EX

2.2.1 Les caractères actifs

Les règles du langage L^AT_EX se centrent sur l'usage de caractères spécifiques dit « caractères actifs ».

➔Réfs
[13]15,53
[16]—
[31]14

Caractères	Utilisation
<code>\</code>	Il indique le début d'un nom de commande : L ^A T _E X exécute alors les tâches associées au nom de la commande.
<code>{ }</code>	Toujours par couple, elles encadrent et isolent des ensembles spécifiques, d'où leur nom de délimiteur .
<code>%</code>	Il indique que ce qui le suit est un commentaire. Jusqu'au prochain passage à la ligne, L ^A T _E X ignore ce qui est écrit et ne l'affiche pas.
<code>\$ _ ^</code>	Ils participent à la rédaction de mathématiques. Voir page 38.
<code>&</code>	Il participe à la constitution des tableaux. Voir page 47.
<code>#</code>	Il participe à la définition de nouvelles commandes. Voir page 58.
espace	Plusieurs espaces consécutives valent une seule espace.
retour à la ligne	Un retour à la ligne est considéré comme une espace. Deux retours à la ligne consécutifs ou plus comme un seul et unique changement de paragraphe.
<code>@</code>	Très particulier, il peut être activé pour devenir un caractère utilisable dans les noms de commandes en programmation avancée, ceci avec l'utilisation de <code>\makeatother</code> (activation) et de <code>\makeatletter</code> (désactivation).

TABLE 2.1 – Caractères actifs

En programmation avancée, ces caractères peuvent être remplacés par d'autres ou redéfinis pour se voir attribuer d'autres rôles. Qui serait intéressé par ce point doit faire une recherche sur la notion de **catcode**, ou code de catégorie (de caractère)⁶ profondément lié à la manière dont L^AT_EX lit un document.

L'affichage de tels caractères actifs dans le texte final (ne serait-ce par exemple que pour les citer dans le tableau ci-dessus) s'obtient par le recours à des commandes présentées dans la table 4.3, page 17.

5. Dans toute la suite, les paquets seront indiqués avec ce code visuel.

6. Par exemple : <http://www.math.u-psud.fr/~bernardofpc/ens/CIES/Avance.pdf>.

2.2.2 Les groupes

Pour que L^AT_EX puisse présenter un endroit spécifique d'un document, il lui faut savoir quel est l'état des différentes valeurs des paramètres de présentation : fonte à utiliser, format de la page et ainsi de suite. Pour cela est utilisée la notion de **groupe**. Un groupe est au sens le plus large une zone de notre document. Un groupe peut être délimité de deux façons :

- par des accolades ouvrantes et fermantes ;
- par des crochets ouvrants et fermants quand ils servent à indiquer un argument d'une commande ;
- par des commandes d'environnement, telles `\begin{document}` et `\end{document}`, présentées ci-après.

Un groupe peut contenir d'autres groupes (des sous-groupes) et ainsi de suite. En terme d'état, chaque sous-groupe récupère l'état du groupe au moment où il est ouvert. Cet état initial peut ensuite être librement modifié dans le sous-groupe. Cependant, lorsqu'un sous-groupe s'achève et que L^AT_EX revient dans le groupe, il reprend l'état qu'avait le groupe, indépendamment de tous les changements apportés par le sous-groupe.

L'exemple ci-dessous montre ce principe avec deux commandes opérant un changement de présentation du texte à partir du moment où elles apparaissent dans le texte : `\itshape` fait passer le texte en italique, `\upshape` le fait passer en caractères droits (dits romains).

```
Voici un {exemple \itshape de {\upshape texte illustrant les} groupes}
sous \LaTeX.
```

Voici un exemple *de* texte illustrant les *groupes* sous L^AT_EX.

CODE 2.2 – Exemple de groupe

Ici, dans le groupe initial, L^AT_EX rédige en caractères romains. Lorsqu'il passe dans le sous-groupe, il garde ce réglage jusqu'à rencontrer `\itshape` qui le fait passer à l'italique. Lorsqu'il passe dans le sous-sous-groupe, il reçoit immédiatement la consigne de passage en caractères droits. En l'absence d'autres réglages, ce format perdure jusqu'à la fin du sous-sous-groupe. Lorsque L^AT_EX quitte cette zone, il revient au sous-groupe avec sa présentation italique. Le mot « groupes » est donc en italique. De même, une fois qu'il quitte le sous-groupe, il revient au réglage qu'avait le groupe : l'écriture en romain.

2.2.3 Les différentes commandes

L^AT_EX distingue tout d'abord lettres majuscules et minuscules⁷ : la commande `\Test` peut avoir une définition différente de la commande `\test` comme de la commande `\TesT`.

Les commandes sont parfois dotées d'arguments. Ceux-ci sont placés immédiatement après le nom de la commande, sans espace : ils sont le plus souvent introduits par des `{` lorsqu'ils sont obligatoires, par des `[`⁸ lorsqu'ils sont facultatifs et conclus respectivement par des `}` et des `]`. De fait, un argument est un groupe.

Trois grands types de commandes coexistent :

- la commande simple avec arguments sous la forme usuelle `\macro{arg1}...{argN}`. Elle exécute normalement une action à l'endroit où elle se trouve dans le texte et agit sur ou avec ses arguments. Par exemple, `\textbf{texte}` mettra le texte en argument en gras. Les arguments ont des tailles limitées à un paragraphe au plus.
- la commande **bascule**, le plus souvent sans argument. Celle-ci modifie le comportement de L^AT_EX jusqu'à la fin du groupe dans lequel elle se situe. Par exemple, `\bfseries` fait que le texte qui suit s'écrit désormais en gras.

7. L^AT_EX est dit « sensible à la casse ».

8. Les crochets ne sont pas considérés comme des caractères actifs, contrairement aux accolades.

- la commande **environnement** de la forme `\begin{environnement}` toujours suivie plus loin de la nécessaire fin⁹ `\end{environnement}`. La zone délimitée par ces deux commandes, qui peut contenir de nombreux paragraphes, est alors modifiée. Ainsi, l'environnement `center` va centrer tout le texte placé entre les deux commandes.

Dans la suite, ces différentes variantes seront respectivement appelées commandes, bascules et environnements.

2.2.4 Unités de mesure

Parmi les arguments qui peuvent être communiqués à des commandes \LaTeX se trouve des longueurs. Ces commandes permettent le plus souvent de faire des réglages, par exemple insérer à la fin de cette phrase une espace de 2cm .

Les unités de longueur absolues connues par \LaTeX sont :

Unité	Nom	Définition	Mesure (en mm)
cm	centimètre	1 cm = 0,01 m	10 mm
mm	millimètre	1 mm = 0,1 cm	1 mm
in	pouce (<i>inch</i>)	1 in = 2,54 cm	25,4 mm
bp	gros point (<i>big point</i>)	72 bp = 1 in	0,3527778 mm
pt	point	72,27 pt = 1 in	0,3514598 mm
pc	pica	1 pc = 12 pt	4,2175176 mm
dd	point didot	1157 dd = 1238 pt	0,3760650 mm
cc	cicéro	1 cc = 12 dd	4,5127803 mm
sp	point d'échelle (<i>scaled point</i>)	65536 sp = 1 pt	0,0000054 mm

TABLE 2.2 – Unités de longueur absolues

Par ailleurs, \LaTeX peut manipuler des longueurs relatives, autrement dit des longueurs qui dépendent de la taille de certains caractères de la police courante. Il existe pour cela deux unités :

Unité	Nom
em	cadratin (largeur de la lettre M)
ex	hauteur d'x

TABLE 2.3 – Unités de longueur relatives

Pour être complet, \LaTeX admet également des longueurs élastiques¹⁰ qui se présentent à titre d'exemple sous la forme suivante : `1cm plus 1ex minus 2mm`. Dans cet exemple, la longueur est comprise entre 1cm–2mm et 1cm+1ex selon les contraintes présentes ; 1cm est la longueur en l'absence de contraintes.

2.3 La compilation

Pour obtenir le document mis en forme, il faut effectuer une **compilation** du fichier `.tex`, autrement dit le faire lire et retranscrire par \LaTeX en un document d'un autre format qui nous donnera l'affichage souhaité, généralement un fichier `.dvi` ou un fichier `.pdf`.

➔Réfs
[13]19–30
[16]—
[31]7–12

2.3.1 La mécanique de la compilation

Lors de la compilation, \LaTeX analyse et transforme les différentes commandes en une suite de fonctions fondamentales qui lui dictent la manière de présenter les pages de texte. Au fur et à

9. Sous peine d'obtenir une erreur de \LaTeX !

10. Elles ne sont qu'évoquées ici mais sont largement développées dans le cours avancé de Manuel PÉGOURIÉ-GONNARD [30]. Ces définitions sont également vues dans l'annexe C en ligne de [9] : <http://latex-pearson.org/ressources/2010/annexe-C.pdf>.

mesure, selon les règles usuelles et les modifications apportées parfois par les commandes, \LaTeX prépare des boîtes contenant des caractères puis des boîtes contenant des boîtes de caractères autrement dit des boîtes-mots, puis des boîtes-lignes puis des boîtes-paragraphe. À chaque étape, il prépare également les espaces entre les boîtes selon des règles typographiques¹¹ ou des règles imposées par les commandes.

Cet empilement de boîtes finit par générer une page. Une fois une page générée, \LaTeX passe à la suivante et ne revient pas en arrière.

De ce fait, \LaTeX ne peut créer directement des éléments demandant une vision de tout le document : une table des matières, un index ou un renvoi vers une page. Pour obtenir de tels éléments, il faudra compiler deux ou trois fois le document, \LaTeX réutilisant des informations stockées par la dernière compilation dans des fichiers annexes prévus pour ce type de fonctionnalités.

2.3.2 Les fichiers générés par la compilation

La compilation produit plusieurs fichiers, chacun ayant un rôle spécifique pour l'utilisateur et pour \LaTeX :

- le fichier `.dvi`¹², un fichier qui restitue un document finalisé.
- le fichier `.log`, ou **journal**. Il contient des informations sur le travail effectué lors de la compilation.
- le fichier `.aux`, ou « auxiliaire ». Il contient des informations de compilation que \LaTeX réutilise lors de la compilation suivante. Parmi celles-ci, se trouvent des informations sur les références dans le document, des numéros de pages, des éléments servant à la table des matières.
- et parfois d'autres fichiers en cas d'utilisation de commandes ou de paquets spécifiques. Par exemple, le paquet `minitoc` génère de nombreux fichiers `.mtc` (suivis de chiffres).

2.4 La gestion des erreurs

L'erreur est hélas humaine... et \LaTeX déteste les erreurs. Les erreurs interfèrent avec la compilation et génèrent souvent une restitution partielle du fichier `.dvi`. \LaTeX peut en effet s'arrêter en cours de compilation si les erreurs sont trop importantes.

Dans ces cas-là, il est recommandé de regarder le fichier `journal`. Ce fichier contient souvent l'information de la ligne où est survenue l'erreur et assez fréquemment sa raison : une commande inconnue, une accolade oubliée, des mathématiques mal tapées. Ce fichier est donc d'une aide précieuse, quoique rédigé en anglais. La plupart des éditeurs \LaTeX permettent d'ailleurs de visionner ce fichier lors de la compilation pour comprendre et corriger les erreurs.

Pour sauver le commun des mortels de certaines erreurs assez complexes, l'annexe E a été constituée sur ce sujet, page 107.

➤Réfs
[13]29–30
[16]—
[31]187–200

2.5 L'impression

Le fichier `.dvi`, s'il peut être visualisé, ne peut être imprimé directement. Il faut le convertir en un autre format et, pour cela, utiliser un programme extérieur à \LaTeX . De façon courante, il est converti à l'aide du programme `dvips` en document postscript (dit fichier `.ps`) lisible et imprimable. Ceci permettra d'utiliser certains paquets spécifiques présentés en section 9.1. Qui plus est, un document `.ps` peut être converti en `.pdf`. Un second programme de conversion est ici utilisé : `ps2pdf`.

➤Réfs
[13]22
[16]—
[31]12–14

11. Entre autres, la règle de césure qui distingue \LaTeX d'autres logiciels.

12. DVI signifie « DeVice-Indépendant » soit « indépendant du type d'unité ou du périphérique ».

2.6 Des compilations alternatives

La compilation décrite jusqu'ici de base sur la mécanique historique. Les variantes de \LaTeX ont revu cette mécanique et l'ont simplifié.

Ainsi, il est possible d'utiliser le programme `pdflatex` qui génère directement un fichier `.pdf` à partir du fichier `.tex`. S'il présente des incompatibilités avec certaines des fonctionnalités vues dans la suite de ce cours, essentiellement le paquet `pstricks`, ce programme autorise en contrepartie l'insertion des images `.jpg`.

Il est également possible de retenir pour la compilation les programme `xetex` ou `lualatex`. Ils permettent de la même manière de passer du fichier `.tex` au fichier `.pdf`. Toutefois, ces programmes demandent d'appliquer un encodage différent au fichier source et il implique de placer un préambule différent de celui utilisés normalement avec \LaTeX . Ces points sont présentés en page 81.

La figure suivante résume l'ensemble de ces processus.

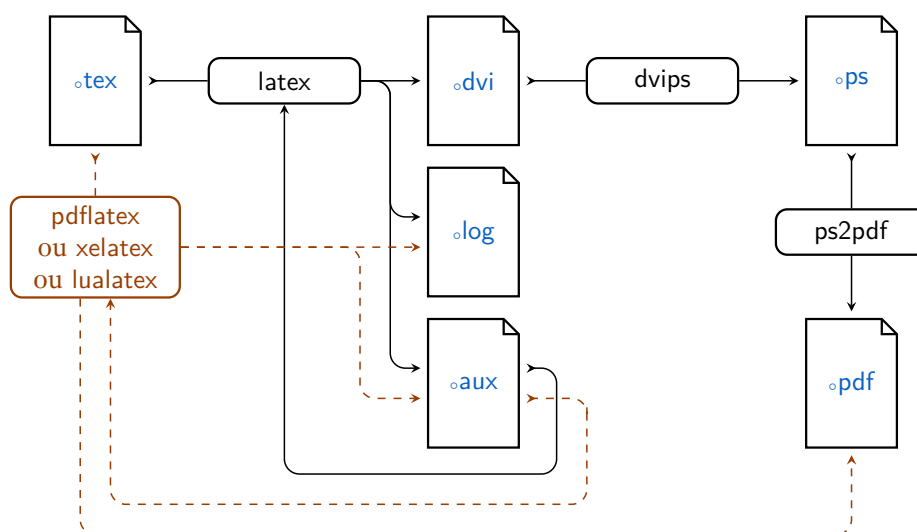


FIGURE 2.1 – Du fichier `.tex` au fichier `.pdf`

Deuxième partie

Notions courantes

Chapitre 3

Préambule et classe du document

3.1 Un préambule minimal

Le préambule varie selon la chaîne de compilation qui a été choisie.

➔Réfs
[13]31–36
[16]17–20
[31]39–41

3.1.1 Cas avec \LaTeX et \pdfLaTeX

Un document en français devrait toujours contenir au moins les commandes suivantes :

```
\documentclass[options]{classe}  
\usepackage[frenchb]{babel}  
\usepackage[T1]{fontenc}  
\usepackage[latin1]{inputenc}  
\usepackage{lmodern}  
\begin{document}  
  
\end{document}
```

CODE 3.1 – Un code minimal pour \LaTeX

Le chargement des paquets `fontenc` et `inputenc` font appel à la notion d’encodage. Nous nous contenterons d’indiquer que l’appel du paquet `fontenc` avec l’option `T1` permet à \LaTeX d’afficher des lettres accentuées, non chargées par défaut. L’appel du paquet `inputenc` indique à \LaTeX l’encodage de notre texte, que ce soit `latin1`, `utf8` ou `applemac`.

3.1.2 Cas avec \XeLaTeX et \LuaLaTeX

Par rapport au cas précédent, plusieurs modifications sont nécessaires :

- modifier dans l’éditeur de fichier `.tex` le paramétrage du programme¹ servant à compiler les fichiers `.tex`. Il est à noter que, par défaut, \XeLaTeX génère directement des fichiers `.pdf` à la manière de `pdflatex` ;
- mettre son fichier en encodage UTF8 afin qu’il soit compilable ;
- retrancher les paquets et fonctionnalités incompatibles. Ainsi, `inputenc` et `fontenc` ne doivent pas être utilisés avec \XeLaTeX et \LuaLaTeX : ils sont à remplacer par `fontspec` qui gère les polices de caractères.

Le préambule d’un document devrait donc contenir au minimum les lignes suivantes :

1. Pour Texmaker. Il faut aller dans le menu `Options` puis `Configurer Texmaker`. Dans la ligne `LaTeX`, il faut remplacer le terme `latex` par `xelatex` ou `lualatex`.

```

\documentclass[options]{classe}
\usepackage[frenchb]{babel}
\usepackage{fontspec}
\defaultfontfeatures{Ligatures=TeX}

\begin{document}

\end{document}

```

CODE 3.2 – Un code minimal pour Xe_{La}T_EX et Lua_{La}T_EX

La commande `\defaultfontfeatures` et l’option précisée ci-dessus permettent un réglage minimal conservant les ligatures classiques de \LaTeX . Ainsi, `---` sera bien converti en `—`.

Quelques précisions sur l’usage de Xe_{La}T_EX sont données en page 81.

3.2 La classe du document

3.2.1 Les grands classiques

Il reste dans les cas ci-dessus à choisir la *classe* et les *options*. La **classe du document** impacte la plus grande part de la présentation et de la structure du document. Ainsi, un livre et une lettre ne demandent pas les mêmes règles de présentation. La classe se choisit principalement parmi les suivantes : `article`, `book`, `report` et `letter`.

Sans options, \LaTeX va retenir une présentation par défaut de la classe. La composition peut être modifiée en indiquant ses options, séparées par des virgules. Les options principales de la composition sont :

- le **corps**, taille standard des caractères : `10pt`, `11pt` ou `12pt`. Sous \LaTeX , toutes les tailles des caractères sont définies relativement à cette taille standard. La changer là fait ainsi s’adapter toute la présentation du document.
- la composition des pages en recto ou recto-verso : `oneside`, `twoside` ;
- le format A4, A5 : `a4paper`, `a5paper` ;
- le format paysage : `landscape` ;
- la présence d’une ou deux colonnes par page : `onecolumn` et `twocolumn` ;
- la position des équations : centrée par défaut, à gauche avec `fleqn` ;
- la position de la numérotation des équations : à droite par défaut, à gauche avec `leqno`.
- le mode brouillon avec `draft` : les images ne sont pas affichées et les débordements de ligne indiqués. Voir page 20.

Les options placées dans la classe du document ont la particularité d’être reprises dans les appels des différents paquets chargés à la suite de cette première commande. Ainsi, l’option `draft`, pour le paquet `hyperref`, désactive l’ensemble des liens hypertextes.

3.2.2 Pour aller plus loin

Les classes citées ci-avant ne sont qu’une petite partie des classes existantes, tout au mieux un ensemble de base. D’autres classes existent pour répondre à de nombreux autres besoins. En voici quelques unes :

- les classes *KOMA-script*² — `scrartcl`, `scrbook`, `scrreprt` et `scrllttr2` — qui retranscrivent dans des formats plus européens les classes mentionnées plus haut ;
- `lettre` permettant de rédiger une lettre en français ;
- `memoir` qui permet de rédiger des mémoires de fin d’études et qui, entre autres, permet d’accéder à d’autres tailles de caractères par défaut ;

2. Bertrand MASSON [25] décrit en français les avantages de ces classes sur les classes traditionnelles sur son ancien site : <http://bertrandmasson.free.fr/index.php?categorie5/latex-koma-script>

- `moderncv` permettant de rédiger un CV, présentée en annexe C ;
- `beamer` permettant de faire une présentation.³

3.3 Les autres paquets usuels

Le paquet `babel` avec l'option `frenchb`, fait des réglages associés à la typographie et la langue française, \LaTeX étant nativement anglophone. Entre autres, les énumérations se font avec des tirets, non des points ; les grandes parties du document portent des noms français et, par exemple, « chapter » devient « chapitre », « Table of contents » devient « Table des matières ». Le paquet `babel` donne également accès à quelques commandes complémentaires présentées en table 4.4.

Le paquet `lmodern` modifie légèrement la police de caractère utilisée (par le chargement d'une version bien plus récente) à l'affichage du document `.pdf`. Elle est beaucoup plus lisible et régulière à l'écran.

3. Ce point devrait être traité dans ce document à l'avenir. Divers sites présentent cependant cette classe. Par exemple <http://mcclnews.free.fr/latex/introbeamer.php> qui catalogue en prime une grande variété de thèmes pour `beamer`.

Chapitre 4

Le texte

TeX effectue automatiquement de très nombreux réglages sur le texte. Le principal mais aussi le moins visible pour les personnes peu familières des règles typographiques est le respect des règles d'espacement, que ce soit entre mots et signes de ponctuation ou entre mots eux-mêmes. TeX peut ainsi recourir à la césure pour que les espacements entre les mots soient de taille à peu près régulière tout au long du texte. En cela il est considéré comme un traitement de texte.

Toutefois, de nombreuses choses restent à la main de l'utilisateur afin de faire ressortir la logique du texte. Elles sont évoquées dans ce chapitre.

4.1 Les caractères

4.1.1 La police de caractère

TeX compose les textes par défaut en **police de caractère** *Computer Modern*. Changer ce comportement peut être relativement complexe car il faut alors définir à TeX de multiples réglages pour qu'il puisse correctement gérer les polices de caractères souhaitées et parfois même créer des fichiers spécifiques.

Aussi, plutôt que de rentrer dans le détail de l'utilisation de différentes polices¹, il est présenté ici une méthode plus limitée mais simple d'utilisation : utiliser des paquets chargeant les polices². Certains paquets permettent de charger des polices pour le texte et les maths :

- `mathptmx` pour *Times* ;
- `fourier` pour *Utopia*.

D'autres paquets doivent être accompagnés de paquets secondaires pour charger des polices mathématiques adaptées :

- `palatino` et `euler` (maths) pour *Palatino* ;
- `bookman`, `kmath` (maths) et `kerkis` (maths) pour *Bookman* ;
- `newcent` et `fouriernc` (maths) pour *New Century Schoolbook*.

D'autres enfin ne livrent qu'une police de caractères textuelle :

- `chancery` pour *Zapf Chancery* ;
- `charter` pour *Charter* ;
- `concrete` pour *Concrete*³.

1. Voir par exemple <<http://zoonek.free.fr/LaTeX/Fontes/fontes.html>> de Vincent ZOONEKYND.

2. Voir <http://benoit.rivet.free.fr/tex/tex_polices_exemples.htm> de Benoit RIVET, <<http://www.cuk.ch/articles/4237>> de Franck PASTOR[28] ou bien encore <<http://web.eecs.utk.edu/~mgates3/docs/latex-fonts.pdf>>.

3. Autre police de caractères faite par Donald E. KNUTH.

4.1.2 Le style des caractères

La notion de style des caractères est un peu trop vague pour être utilisable en typographie. Ainsi, sous \LaTeX , au sein d'une police de caractère, une fonte de caractères est définie par son encodage, sa **famille**, sa **forme**, sa **série**⁴ et sa taille. En considérant pour simplifier la question de l'encodage traitée par notre préambule de document, quatre paramètres peuvent donc être manipulés à loisir. Les trois premiers se définissent ainsi :

- famille : elle distingue les fontes présentant ou pas des **empattements** ainsi que des fontes présentant une **chasse fixe** ou pas. En typographie classique, chacune de ces familles serait une police de caractère à part entière ;
- forme : elle distingue les fontes droites, italiques, penchées, italiques droites mais aussi les fontes à petites capitales ;
- série : épaisseur (ou graisse) et étroitesse de la fonte.

La famille, la forme et la série

Par défaut, \LaTeX utilise la police de caractère « *Computer Modern* » et nous place dans la famille romaine, forme droite et graisse moyenne.

Plusieurs commandes⁵ permettent de modifier ces paramètres et sont listées dans la table suivante.

Sélection	Commande	Environnement	Bascule	Bascule historique
Famille romaine	<code>\textrm{texte}</code>	<code>rmfamily</code>	<code>\rmfamily</code>	<code>\rm</code>
Famille sans empattement	<code>\textsf{texte}</code>	<code>sffamily</code>	<code>\sffamily</code>	<code>\sf</code>
Famille à chasse fixe	<code>\texttt{texte}</code>	<code>ttfamily</code>	<code>\ttfamily</code>	<code>\tt</code>
Forme droite	<code>\textup{texte}</code>	<code>upshape</code>	<code>\upshape</code>	
Forme penchée	<code>\textsl{texte}</code>	<code>slshape</code>	<code>\slshape</code>	<code>\sl</code>
Forme italique	<code>\textit{texte}</code>	<code>itshape</code>	<code>\itshape</code>	<code>\it</code>
FORME PETITES CAPITALES	<code>\textsc{texte}</code>	<code>scshape</code>	<code>\scshape</code>	<code>\sc</code>
Série moyenne	<code>\textmd{texte}</code>	<code>mdseries</code>	<code>\mdseries</code>	
Série grasse	<code>\textbf{texte}</code>	<code>bfseries</code>	<code>\bfseries</code>	<code>\bf</code>

TABLE 4.1 – Différents styles de caractère

Certaines combinaisons peuvent ne pas être disponibles selon la police de caractère utilisée pour un texte. Ainsi, pour la *Computer Modern*, les petites capitales n'existent pas en italique.

La commande `\emph{texte}` permet de gérer un cas particulier d'italique. L'italique sert normalement à attirer l'attention dans un texte composé en romain. Toutefois, dans un passage en italique, pour attirer l'attention, il faut revenir au romain, « italique de l'italique ». La commande `\emph` gère ce comportement spécifique à la différence de `\textit`, ce qui explique qu'elle soit plus couramment employée.

La taille des caractères

Relativement au corps fixé lors de la définition de la classe, la taille du texte peut être modifiée par différentes commandes citées dans le tableau suivant de la plus petite à la plus grande : `\tiny` donne ainsi des caractères plus petits que `\scriptsize`.

4. Les noms des commandes vues ici indiquent d'ailleurs ce qu'elles modifient parmi ces trois éléments : `..family`, `..shape` ou `..series`.

5. Les bascules à deux caractères présentées dans cette table sont considérées comme désuètes car moins flexibles que les autres bascules citées ici : elles ne permettent pas d'obtenir un texte en italique gras par exemple.

Caractères	Commandes
Petits	<code>\tiny \scriptsize \footnotesize \small</code>
Moyens	<code>\normalsize</code>
Grands	<code>\large \Large \LARGE \huge \Huge</code>

TABLE 4.2 – Différentes tailles de caractères

```
\tiny Ceci \scriptsize est \footnotesize un \small petit \normalsize exemple \large de
\Large ce \LARGE qui \huge est \Huge faisable.
```

Ceci est un petit exemple de ce qui est faisable.

CODE 4.1 – Tailles de caractères

4.1.3 Les caractères spéciaux

Si la plupart des caractères s'obtiennent directement par les touches du clavier, quelques uns, plus rares, peuvent être obtenus par des commandes, tout particulièrement les voyelles accentuées majuscules. Le tableau 4.3 ainsi que le tableau 2.1 des caractères actifs résument les principales commandes pour un texte en français.

Caractère	Commande	Caractère	Commande
Accent aigu : é	<code>\' {lettre}</code>	Accent grave : è	<code>\` {lettre}</code>
Accent circonflexe : ê	<code>\^ {lettre}</code>	Tréma : ë	<code>\" {lettre}</code>
Cédille : ç	<code>\c {lettre}</code>	E dans l'o : Œ, œ	<code>\OE, \oe</code>
Esperluette : &	<code>\&</code>	Pourcentage : %	<code>\%</code>
Contre-oblique : \	<code>\textbackslash</code>	Dièse : #	<code>\#</code>
Accolage ouvrante : {	<code>\{</code>	Accolage fermante : }	<code>\}</code>
Tiret bas : _	<code>_</code>	Dollar : \$	<code>\\$</code>
Trait d'union : -	<code>-</code>	Intervalle numérique : –	<code>--</code>
Tiret cadratin : —	<code>---</code>	Signe moins : —	<code>-\$-\$</code>
Pied-de-mouche : ¶	<code>\P</code>	Paragraphe : §	<code>\S</code>
Obèle : †	<code>\dag</code>	Double obèle : ‡	<code>\ddag</code>
Copyright : ©	<code>\textcopyright</code>	Trade-mark : ™	<code>\texttrademark</code>
Marque : ®	<code>\textregistered</code>	Espace apparente : □	<code>\textvisiblespace</code>

TABLE 4.3 – Caractères particuliers

L'accent aigu est celui obtenu avec la touche 4 du clavier, l'accent grave, celui avec la touche 7. Sans lettre dans leur argument, les accents apparaissent seuls. Par ailleurs, comme vu précédemment, le symbole de l'euro s'obtient en chargeant le paquet `eurosym`.

Dans le cas particulier du placement d'un accent sur la lettre « i », il faut utiliser en lieu et place du « i » la commande `\i` qui permet d'afficher un i sans point, laissant la place pour l'accent souhaité.

Il faut noter ici que la langue française demande à ce que les majuscules soient accentuées⁶.

Le chargement du paquet `babel` avec l'option `frenchb` met à disposition quelques commandes d'usage courant.

6. Sans cela, il est par exemple impossible de dire si « LES RETRAITES » désigne les retraites ou les retraités.

Caractère	Commande	Caractère	Commande
Guillemets ouvrants : «	<code>\og</code>	Guillemets fermants : »	<code>\fg</code>
n ^o	<code>\no</code>	N ^o	<code>\No</code>
n ^{os}	<code>\nos</code>	N ^{os}	<code>\Nos</code>
1 ^{er}	<code>1\ier</code>	1 ^{ers}	<code>1\iers</code>
1 ^{re}	<code>1\iere</code>	1 ^{res}	<code>1\ieres</code>
2 ^e	<code>2\ieme</code>	2 ^{es}	<code>2\iemes</code>
1 ^o	<code>\primo</code>	2 ^o	<code>\secundo</code>
3 ^o	<code>\tertio</code>	4 ^o	<code>\quarto</code>
10 ^o	<code>10\degre</code>	M ^{me}	<code>M\up{me}</code>

TABLE 4.4 – Caractères particuliers issus de `babel`

Pour éviter une erreur courante, « Monsieur » s’abrège en « M. ».

Enfin, parmi d’autres, le paquet `textcomp` met à disposition quelques caractères spéciaux parfois utiles.

Caractère	Commande	Caractère	Commande
%o	<code>\textperthousand</code>	※	<code>\textreferencemark</code>
o	<code>\textopenbullet</code>	•	<code>\textbullet</code>

TABLE 4.5 – Caractères particuliers issus de `textcomp`

Le paquet `pifont` permet d’accéder à la police de caractère *Zapf Dingbats* dont chaque caractère est un **dingbat** (ou **casseau**), autrement dit un dessin accessible comme un caractère. La table des caractères est la suivante :

31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110
111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130
131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150
151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170
171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190
191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210
211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230
231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250
251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270

TABLE 4.6 – Table de caractère associée à `pifont`

Quelques commandes permettent, sur la base du numéro associé à chaque caractère dans cette

table, d'utiliser ces caractères. `\ding` pour un caractère isolé, `\dingfill` pour un remplissage par un caractère, `\dingline` pour une ligne de caractère avec marges gauche et droite.

```
Une feuille : \ding{166}. Puis \dingfill{217} pour remplir la ligne affichée.
Et une ligne seule : \dingline{73}
```

Une feuille : ☼. Puis → → → pour remplir la ligne affichée. Et une ligne seule :
 ☆ ☆ ☆ ☆ ☆ ☆ ☆ ☆ ☆ ☆ ☆ ☆ ☆ ☆ ☆ ☆ ☆ ☆ ☆

CODE 4.2 – Commandes issues de `pifont`

Avec `pifont`, deux environnements de liste sont également disponibles et mentionnés page 27.

4.1.4 Pour aller plus loin

La gestion des polices de caractères est grandement simplifiée avec \LaTeX . Ce sujet est présenté au chapitre 10, page 81.

4.2 Autour des caractères

4.2.1 Les espaces

\LaTeX ne prend pas en compte les espaces ou retours à la ligne multiples présents dans le fichier `.tex`. Des commandes sont en effet dédiées à ce point⁷ et en voici quelques unes réparties entre espaces horizontales et espaces verticaux⁸.

Les espaces horizontales

Espace	Commande
Espace simple	<code>_</code> (une espace)
Indentation	<code>\indent</code>
Suppression d'indentation	<code>\noindent</code>
Espace horizontale fixée	<code>\hspace{dimension}</code>
Espace horizontale fixée impérative	<code>\hspace*{dimension}</code>
Ressort horizontal	<code>\hfill</code>
Correction d'italique	<code>\/</code>

TABLE 4.7 – Espacements horizontaux

L'**indentation** est l'espace horizontale qui, en français, commence la première ligne d'un paragraphe. Il est proposé par défaut mais peut être retranché en plaçant la commande `\noindent` en début de paragraphe, de même qu'il peut être forcé avec `\indent`.

La commande `\hspace` présente deux variantes. La version non étoilée est considérée comme une espace qui est ignorée si elle est présente en début de ligne ou en fin de ligne dans un paragraphe. La version étoilée est par contre impérative.

Le ressort horizontal est un espace qui prend toute la place disponible. Plusieurs ressorts placés sur la même ligne auront chacun la même dimension, comme l'illustre l'exemple suivant.

La correction d'italique est une espace fine permettant d'éviter que le passage de caractères italiques à romains ne fassent se croiser des caractères hauts. Elle est indiquée ici car, si elle est

7. La logique voudrait qu'elles soient utilisées en phase de finalisation du document, un saut de page pouvant par exemple devenir inutile ou mal placé si le document évolue toujours.

8. Espace est féminin lorsque ce mot désigne un blanc entre deux mots.


```
À n'en pas douter, le major Alexandre Van Otenberg agissait de manière schtroumpfement
peu fanfaronne.

À n'en pas douter, le major Alexandre Van Otenberg agissait de manière
schtroumpfe\~ment peu fanfaronne.
.....

À n'en pas douter, le major Alexandre Van Otenberg agissait de manière schtroump-
fement peu fanfaronne.
À n'en pas douter, le major Alexandre Van Otenberg agissait de manière schtroumpfe-
ment peu fanfaronne.
```

CODE 4.4 – Césure dans le texte

Un réglage de césure peut s'opérer dans tout le document en plaçant la commande suivante dans le préambule :

```
\hyphenation{schtroump-fe-ment}
```

CODE 4.5 – Césure en préambule

L'espace insécable `~` permet d'obtenir l'effet inverse : \LaTeX ne pourra pas couper la ligne sur cette espace. Ceci permet d'empêcher par exemple de séparer un nom d'un prénom, un nombre de son unité, le mot « page » du numéro qui le suit.

4.2.3 Le soulignement

Les pratiques typographiques considère que le soulignement ne devrait être utilisé que si l'italique n'est pas disponible sur son traitement de texte. Ce qui suit devrait donc être utilisé de façon assez exceptionnelle.

La commande de soulignement par défaut est `\underline{texte}`. Elle permet de traiter quelques mots et gère mal le changement de ligne au sein des paragraphes.

Le paquet `ulem` simplifie cette approche. Il est recommandé de l'appeler de la manière suivante :

```
\usepackage[normalem]{ulem}
```

CODE 4.6 – Appel pour le soulignement

Par défaut, ce paquet transforme `\emph` en une commande de soulignement. Pour garder un comportement plus standard de \LaTeX , l'option `normalem` restitue à la `\emph` sa fonction d'origine. Par ailleurs, ce paquet met à disposition la commande `\uline` bien plus efficace pour traiter les paragraphes, de même que quelques variantes un peu plus exotiques.

Soulignement	Commande
<u>Simple</u>	<code>\uline{texte}</code>
<u>Double</u>	<code>\uuline{texte}</code>
<u>Ondulé</u>	<code>\uwave{texte}</code>
Par tiret	<code>\dashuline{texte}</code>
Pointillé	<code>\dotuline{texte}</code>
<u>Barré</u>	<code>\sout{texte}</code>
<u>À double barre</u>	<code>\xout{texte}</code>

TABLE 4.9 – Différents soulignements

4.2.4 L'encadrement

Deux commandes de base permettent de traiter l'encadrement de quelques mots. La première est `\fbox{texte}` qui encadre le texte qui lui est soumis. La seconde permet de gérer la *dimension* horizontale de la boîte et l'alignement du *texte* qu'elle contient :

```
\framebox[dimension][alignement]{texte}
```

CODE 4.7 – Une boîte encadrée

L'*alignement* est à choisir parmi trois valeurs : **l** pour aligner à gauche, **r** pour aligner à droite et **s** pour étirer les espaces pour remplir toute la boîte.

Si les boîtes à traiter dépassent la simple ligne, il faut utiliser des commandes gérant des boîtes-paragraphe pour gérer les changements de ligne. La commande `\parbox` est pensée en ce sens :

```
\parbox{dimension}{texte}
```

CODE 4.8 – Une boîte de texte encadrée

La dimension permet de fixer la largement de la boîte, le texte retournant alors à la ligne automatiquement.

```
Voici un \fbox{cadre} idéal pour faire \framebox[3cm][l]{un exemple de}
\framebox[4cm][s]{nos grands travaux}.
\fbx{\parbox{5cm}{Et voici un cadre plus étroit pour traiter des autres sujets.}}
```

Voici un cadre idéal pour faire un exemple de nos grands travaux.

Et voici un cadre plus étroit pour
traiter des autres sujets.

CODE 4.9 – Quelques encadrements

Pour aller plus loin

Deux paquets, `framed` et `fancybox`, permettent d'obtenir d'autres styles d'encadrement, qu'ils soient à bords arrondis, ombrés ou doublés.

Par ailleurs, \LaTeX est construit pour gérer des boîtes puisque tout son travail revient à les positionner sur la page. D'autres commandes permettent donc de gérer plus avant boîtes et encadrements tel l'environnement `minipage` ou la commande `\savebox`.

4.3 La couleur

La couleur s'obtient avec le paquet `xcolor`¹¹. Certains paquets chargent d'ailleurs automatiquement ce paquet, tel `pstricks`.

4.3.1 La définition des couleurs

La définition de couleurs personnalisées se fait par l'utilisation de l'une des deux commandes suivantes à placer dans le préambule :

➤Réfs
[13]436–440
[16]—
[31]225–227

```
\definecolor{couleur}{système}{composition}
\colorlet{couleur}{formule}
```

CODE 4.10 – Définition de couleur

Le nom de la *couleur* est libre avec pour seule restriction l'absence d'espace dans le nom choisi. Le *système* est, lui, un des systèmes de codage de la couleur possibles. Le tableau ci-dessous précise pour chaque système la façon d'écrire la *composition* d'une couleur selon les composantes du système. Une valeur faible sur une composante marque l'absence de cette composante, une valeur maximale sa présence. En RGB, le noir se code ainsi `0,0,0`.

Système	Composantes	Composition	Exemple
<code>rgb</code>	Rouge, Vert, Bleu (décimal)	<code>[0..1],[0..1],[0..1]</code>	<code>0.82,0.6,0.101</code>
<code>RGB</code>	Rouge, Vert, Bleu (entier)	<code>[0..255],[0..255],[0..255]</code>	<code>209,153,26</code>
<code>HTML</code>	Rouge, Vert, Bleu (hexadécimal)	<code>[00..FF][00..FF][00..FF]</code>	<code>D1991A</code>
<code>cmyk</code>	Cyan, Magenta, Jaune, Noir (décimal)	<code>[0..1],[0..1],[0..1],[0..1]</code>	<code>0,0.268,0.876,0.18</code>

TABLE 4.10 – Systèmes de couleur courants

La commande `\colorlet` permet de définir une couleur par une *formule* utilisant d'autres couleurs déjà existantes. La *formule* est une alternance de noms de couleur et de nombres (des pourcentages) compris entre 1 et 100 et séparés par des « ! » tel, par exemple, `blue!40,blue!40!green` ou bien encore `blue!40!green!75!grey`. La première se comprend comme un mélange de 40% de bleu et de 60% de blanc, la seconde comme un mélange de 40% de bleu et 60% de vert, et la dernière comme la composition de la couleur obtenue avec la seconde expression à 75% et le gris à 25%.

Voici donc deux définitions en pratique :

```
\definecolor{vert}{cmyk}{0.56,0,1,0.27}
\colorlet{vertclair}{vert!60}
```

CODE 4.11 – Quelques couleurs

4.3.2 L'utilisation de couleurs prédéfinies

Le chargement du paquet `xcolor` provoque en fait le chargement de 19 couleurs prédéfinies dont la liste est présentée ci-dessous.

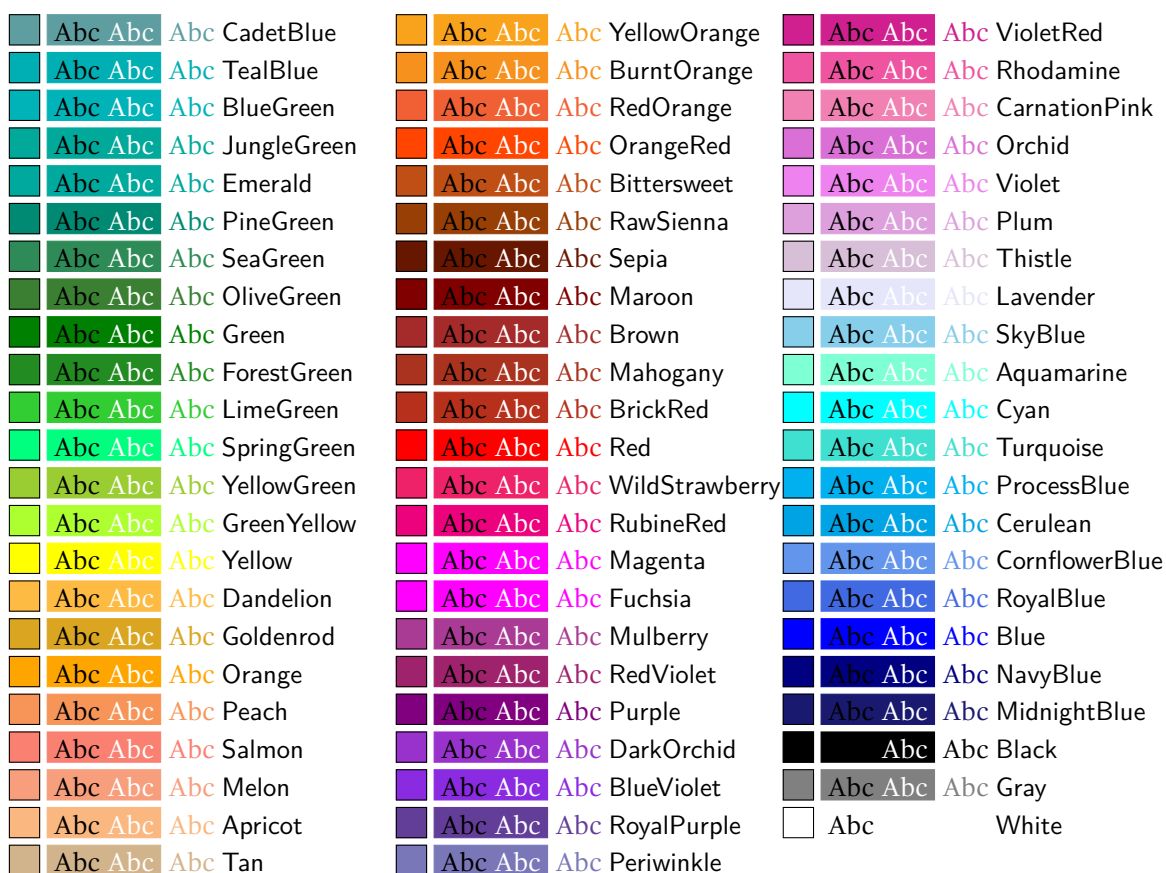
11. En faisant attention à l'ordre de chargement d'autres paquets tels `hyperref`. La documentation de `xcolor` précise ce point.



FIGURE 4.1 – Couleurs de base

Le paquet `xcolor` propose par ailleurs d'élargir cette palette avec 68 couleurs prédéfinies dès qu'il est chargé avec l'option suivante :

```
\usepackage[dvipsnames]{xcolor}
```

FIGURE 4.2 – Couleurs issues de l'option `dvipsnames`

Mais il peut être chargé 151 couleurs nommées en changeant l'option `dvipsnames` en `svgnames` et 317 couleurs nommées en chargeant l'option `x11names`. Dans ce dernier cas, les couleurs sont par lot de 4 variantes en luminosité d'une même teinte. Les noms de ces différentes couleurs sont présentées dans la documentation de `xcolor`.

4.3.3 L'appel des couleurs

Cas du texte

Pour colorer une petite partie de texte avec une *couleur* (en donnant ici le nom tel que défini plus haut), il suffit d'utiliser la commande :

```
\textcolor{couleur}{texte}
```

CODE 4.12 – Texte en couleur

Pour colorer des zones de texte plus grandes qu'un paragraphe, il convient d'utiliser une bascule qui, visuellement, redéfinit la couleur du pinceau utilisée par \LaTeX :

```
\color{couleur}
```

CODE 4.13 – Bascule de couleur

Et, pour revenir à la normale, il faut utiliser `\color{black}`.

Cas des boîtes

Deux commandes permettent de colorer du texte et les boîtes associées à ce texte.

```
\colorbox{couleur fond}{texte}  
\fcolorbox{couleur du bord}{couleur du fond}{texte}
```

CODE 4.14 – Boîtes en couleur

La première place le texte dans une boîte dont le fond est d'une *couleur* donnée, sans bordure. La seconde ajoute la notion de bordure de la boîte. Elle gère la *couleur du bord* en plus de la *couleur du fond*.

Cas des autres éléments

Les couleurs peuvent s'appliquer aussi dans les tableaux (page 51), dans les liens hypertexte (page 28) dans les figures tracées avec \LaTeX (page 71 et suivantes) mais aussi en fond de page avec la bascule :

```
\pagecolor{couleur}
```

CODE 4.15 – Page en couleur

4.4 L'agencement du texte

4.4.1 L'alignement

Quelques environnements permettent de placer le texte : alignement à droite¹², centrage ou alignement à gauche. D'autres environnements permettent d'obtenir des effets un peu plus spécifiques : présenter une citation ou présenter du code informatique « brut ». Dans ce dernier cas,

12. En typographie, il s'agit d'un « fer à droite ».

LaTeX ne réagit plus aux caractères spéciaux et restitue exactement le texte (espaces multiples et retours à la ligne inclus) tant qu'il ne rencontre pas la fin de l'environnement `verbatim`.

Style	Environnement
Alignement à gauche	<code>flushleft</code>
Centrage	<code>center</code>
Alignement à droite	<code>flushright</code>
Citation	<code>quotation</code>
Code informatique	<code>verbatim</code>

TABLE 4.11 – Différents styles de présentation

4.4.2 Les listes

Les principaux environnements pour faire des listes sont donnés dans la table suivante.

Style	Environnement
liste avec tiret	<code>itemize</code>
liste avec numéro	<code>enumerate</code>
liste avec mots	<code>description</code>

TABLE 4.12 – Différents styles de liste

Dans les deux premiers cas, chaque élément de la liste est introduit par la commande `\item`, sans argument. Cette commande va, selon le type de liste demandé, se comporter différemment et afficher un symbole différent. Ceci implique que, sous LaTeX, il n'y a jamais besoin de taper le tiret ou les numéros de liste.

```
Il y a trois types de mathématiciens :
\begin{itemize}
\item les bons;
\item les mauvais.
\end{itemize}
```

```
Il y a trois types de mathématiciens :
— les bons;
— les mauvais.
```

CODE 4.16 – Une liste simple

La commande `\item` peut recevoir un argument optionnel permettant de remplacer le symbole usuel par un autre. Par exemple, `\item[+]` va afficher le symbole « + ». L'environnement `description` a besoin de cette forme particulière pour fonctionner, mettant en gras le terme mis entre crochets. L'exemple ci-dessous illustre ces différents cas ainsi que la possibilité d'insérer une liste dans une autre.

```

Quelques grandes classes de polices de caractères :
\begin{description}
\item[les classiques] comme la \textit{Garamond};
\item[les modernes] comme
\begin{itemize}
\item[$\blacktriangleright$] la \textit{Futura};
\item[$\blacktriangleright$] la \textit{Bodoni};
\end{itemize}
\item[les calligraphiques] comme l'\textit{Optima}.
\end{description}

```

Quelques grandes classes de polices de caractères :

les classiques comme la *Garamond*;

les modernes comme

► la *Futura*;

► la *Bodoni*;

les calligraphiques comme l'*Optima*.

CODE 4.17 – Une liste revisitée

Le paquet `pifont` ajoute deux autres environnements de liste où le symbole d'énumération, précisé par son code numérique en argument, est tiré de la table de caractère. L'environnement `dingautolist` change de caractère à chaque `\item`, en suivant l'ordre de la table.

```

Visuellement, il y a plusieurs types d'énumération :
\begin{dinglist}{70}
\item les classiques;
\item les fantaisistes.
\end{dinglist}
Dans le second cas, il faudra faire attention :
\begin{dingautolist}{202}
\item au public à qui s'adresse le document;
\item à la réelle nécessité d'une énumération fantaisiste.
\end{dingautolist}

```

Visuellement, il y a plusieurs types d'énumération :

◆ les classiques ;

◆ les fantaisistes.

Dans le second cas, il faudra faire attention :

❶ au public à qui s'adresse le document ;

❷ à la réelle nécessité d'une énumération fantaisiste.

CODE 4.18 – Listes issues de `pifont`

4.4.3 Les notes

Les commandes usuelles

Les notes de bas de page sont obtenues par la commande `\footnote{note}` placée à l'endroit du fichier `.tex` où doit figurer le renvoi¹³ de la note, la disposition de la *note* étant ensuite gérée par L^AT_EX (sous un filet en bas de page, en renvoyant au besoin une part de la note sur la page suivante).

13. Voici une note de bas de page avec un renvoi noté 13.

►Réfs
[13] 86–91
[16] 112–131
[31] 42–43

Plus rares, les notes marginales sont générées par la commande `\marginpar{note}`. Un exemple en est donné par le petit bloc « ➤Réfs » apparaissant à droite de ce texte. Les marges étant généralement peu spacieuses, il est recommandé de composer ces notes en changeant la taille des caractères, comme vu précédemment.

Pour aller plus loin

Le paquet `todonotes` permet de positionner dans un texte des annotations en couleur pour marquer des ajouts ou modifications à placer dans le document. Une liste de ces annotations est également affichable.

Dans le même ordre d'idée, le programme `latexdiff`¹⁴ permet d'obtenir l'affichage en format `.pdf` des modifications entre deux versions d'un document \LaTeX .

4.5 Les hyperliens

Le paquet `hyperref` prend en charge les fonctionnalités liées au document `.pdf` final, principalement les liens hypertextes. Le chargement se fait avec la commande suivante :

```
\usepackage[options]{hyperref}
```

CODE 4.19 – Appel du paquet `hyperref`

Le chargement de ce paquet doit être fait avec un peu plus de prudence car il redéfinit des commandes courantes. Il convient donc de le placer en dernier dans l'ordre de chargement des paquets. Par ailleurs, si ce paquet est ajouté après une première compilation, il faut supprimer les fichiers `.lof`, `.lot`, `.toc` et `.mtc`¹⁵, ces derniers pouvant créer une erreur.

À tout moment dans le document, certaines options¹⁶ peuvent être modifiées en retenant une syntaxe similaire avec la bascule `\hypersetup{options}`.

Les différentes *options*, séparées par des virgules, sont sous la forme *option=valeur*. Voici quelques options courantes et leur signification :

- `colorlinks=true` : les liens apparaissent colorés. Si la valeur est `false`, les liens sont alors encadrés en couleur.
- `breaklinks=true` : les liens longs peuvent être sur plusieurs lignes sans que le lien dysfonctionne.
- `hidelinks` (option sans valeur) : les liens ne sont ni encadrés, ni colorés.
- `linkcolor=couleur` affecte la *couleur* aux liens internes au document. De la même manière, les options `filecolor`, `urlcolor` et `citecolor` modifient respectivement la couleur des liens vers un fichier, des liens vers un lien web et des liens allant vers la bibliographie.
- `linkbordercolor=couleur` affecte la *couleur* au cadre pour les liens internes au document. Il existe bien sûr les options `filebordercolor`, `citebordercolor` et `urlbordercolor`.

Voici l'exemple de ce polycopié. Le paquet `hyperref` est chargé sans option mais la déclaration des options est faite à la suite :

```
\hypersetup{urlcolor=orange3,linkcolor=orange3,citecolor=orange3,breaklinks}
```

CODE 4.20 – Exemple de paramétrages

14. La documentation se trouve à l'adresse <https://www.ctan.org/tex-archive/support/latexdiff/doc>. Elle contient en particulier un exemple d'utilisation.

15. Ces dernières sont dues au paquet `minitoc`.

16. La liste des options modifiables n'a apparemment jamais été dressée.

Des options¹⁷ permettent aussi d'affiner le comportement des lecteurs de fichier `.pdf`. Ainsi, l'option `bookmarks=true` permet d'afficher la table des signets lors de l'ouverture du fichier `.pdf` :

Par ailleurs, pour créer un lien hypertexte comme ceux des notes en page ici, il existe la commande `\href{adresse du lien}{texte affiché}`.

4.6 Pour aller plus loin

Certains paquets génèrent des éléments de présentation de texte répondant à des besoins spécifiques :

- `multicol` permet de gérer localement des affichages sur un nombre de colonnes fixé. Un exemple en est donné en page 67 ;
- `listings` permet de mettre en forme du code informatique sur un très large panel de langages ;
- `algorithm2e` permet de mettre en forme des algorithmes ;
- `lettrine` permet de gérer une lettrine en début de paragraphe ;
- `xwatermark` permet de gérer des filigranes — des textes ou images placés en arrière plan (tel « Confidentiel » ou bien encore « Brouillon »).

17. Elles sont décrites par exemple à l'adresse <http://en.wikibooks.org/wiki/LaTeX/Hyperlinks>.

Chapitre 5

Les structures du document

Sommaire

5.1 Les divisions	31
5.1.1 Les divisions non numérotées	31
5.2 La table des matières	31
5.2.1 La profondeur de la table des matières	31
5.2.2 L'ajout en table des matières	32
5.2.3 Les mini-tables des matières	32
5.2.4 Pour aller plus loin	32
5.3 Les références	33
5.3.1 Les commandes usuelles	33
5.3.2 Pour aller plus loin	33
5.4 L'index	33
5.4.1 Le programme externe makeindex	33
5.4.2 Quelques compléments sur l'indexation	34
5.4.3 Pour aller plus loin	34
5.5 La bibliographie	34
5.5.1 Le fichier bibliographique	35
5.5.2 Le code <code>\Bibliography</code>	35
5.5.3 Le contenu final	35
5.5.4 <code>\Bibliography</code>	36
5.5.5 Pour aller plus loin	36
5.6 Les en-têtes et pieds de page	36
5.6.1 Les commandes usuelles	36
5.6.2 Pour aller plus loin	36
5.7 Le titre	36
5.7.1 Les commandes usuelles	36
5.7.2 Pour aller plus loin	37

Structurer et présenter un texte avec \LaTeX n'est pas de prime abord intuitif : selon le besoin, la méthode diffère. Générer une bibliographie ou constituer un index demande l'utilisation de commandes ou de paquets dédiés. Il faut alors souvent placer des commandes spécifiques à plusieurs endroits dans le document `.tex`. Dans certains cas, pour des restitutions élaborées, il faut avoir recours à des programmes différents de \LaTeX (voir page 33).

Dans la mesure du possible, et à titre d'exemple, ces structures sont utilisées dans ce polycopié, quitte à le surcharger quelque peu.

5.1 Les divisions

Selon la classe de document choisie, des divisions de textes sont disponibles ou pas. Dans les classes usuelles, seule `article` ne reconnaît pas la division chapitre. Les noms de division pour les classes de base sont hiérarchiquement :

»Réfs
[13]32–34,487–501
[16]24–47
[31]40–41

Division	Commande	Niveau
Partie	<code>\part{}</code>	-1
Chapitre	<code>\chapter{}</code>	0
Section	<code>\section{}</code>	1
Sous-section	<code>\subsection{}</code>	2
Sous-sous-section	<code>\subsubsection{}</code>	3
Paragraphe	<code>\paragraph{}</code>	4
Sous-paragraphe	<code>\subparagraph{}</code>	5

TABLE 5.1 – Divisions d’un document

Ces divisions structurent fortement le document : elles changent la présentation des titres, créent des numérotations, génèrent des espacements spécifiques, appliquent des sauts de page.

Ces commandes se tapent toutes ainsi : `\division[titre alternatif]{titre}`. Le *titre alternatif* est optionnel : il remplace le *titre* dans la seule table des matières, permettant ainsi d’y afficher des titres plus courts.

5.1.1 Les divisions non numérotées

Lors de la création d’annexes au document, il est recommandé d’utiliser la bascule `\appendix` qui transforme les numéros de chapitres qui suivent en lettres, en repartant de la lettre A.

Pour ne pas afficher de numérotations dans les divisions comme les chapitres ou sections (typiquement, pour un avant-propos, des remerciements, une préface), il faut utiliser les versions « étoilées » des commandes de division comme `\chapter*{}` ou `\section*{}`. Elles empêchent la numérotation des éléments concernés en gardant leur impact sur la hiérarchie du document.

Il existe par ailleurs des commandes permettant de séparer un document de classe `book` en trois parties, modifiant la numérotation des pages et des chapitres :

- `\frontmatter` pour la partie d’introduction du document avec des numéros de pages en chiffres romains et des chapitres non numérotés ;
- `\mainmatter` pour la partie centrale avec des numéros de page en chiffre arabe et des chapitres numérotés ;
- `\backmatter` qui passe à nouveau en chapitres non numérotés pour la fin du document.

5.2 La table des matières

La table des matières s’obtient avec la commande `\tableofcontents`, placée à l’endroit où elle est souhaitée dans le document.

Cette commande a pour particularité de ne générer une table des matières à jour qu’après deux compilations. Ceci s’explique par la mécanique de compilation de \LaTeX vue en section 2.3 : \LaTeX doit récupérer des informations de la compilation précédente, en l’occurrence le fichier `.toc` qui contient le code \LaTeX présentant le contenu de la table des matières. Ce fichier sera pris en compte lors de la seconde compilation.

»Réfs
[13]181–188
[16]47–68
[31]45–46,273–276

5.2.1 La profondeur de la table des matières

La profondeur de cette table est modifiable avec la commande suivante en préambule :

```
\setcounter{tocdepth}{niveau}
```

CODE 5.1 – Modification de la profondeur de la table des matières

Ce *niveau* correspond à celui indiqué dans le tableau 5.1.

5.2.2 L'ajout en table des matières

L'usage des versions étoilées des divisions peut poser problème car la division n'apparaît alors plus dans la table des matières ! Pour corriger cela, comme cela a été fait pour l'avant-propos de ce document, une commande permet de forcer des ajouts d'éléments dans la table des matières :

```
\addcontentsline{toc}{division}{titre}
```

CODE 5.2 – Ajout d'une entrée à la table des matières

Cette commande se place juste après la commande déclarant la division. La *division* correspond alors au niveau hiérarchique souhaité pour notre ajout dans la table, le plus souvent `chapter` ou `section`. Un exemple est présenté avec le code 5.8.

Cette commande peut en particulier permettre de traiter des éléments manquants par défaut : les index, les bibliographies et la table des matières elle-même. Dans ces derniers cas, toutefois, le paquet `tocbibind`, par son seul chargement, permet de corriger ce point.

5.2.3 Les mini-tables des matières

Afin de faciliter la lecture de document long, il peut parfois être intéressant de mettre à la disposition du lecteur une mini-table des matières, ce que propose le paquet `minitoc`. L'appel de ce paquet dans le préambule demande une option pour spécifier le langage retenu :

```
\usepackage[french]{minitoc}
```

CODE 5.3 – Appel de `minitoc` en français

Juste après `\begin{document}` se place `\dominitoc`. Cette commande demande la récupération dans des fichiers des divisions constituant les mini-tables des matières. Il ne reste alors plus qu'à placer `\minitoc`¹ à l'endroit choisi pour afficher une mini-table².

5.2.4 Pour aller plus loin

La table des matières, la liste des tableaux (voir page 48) et la liste des figures (voir page 53) peuvent être revues en terme de présentation avec le paquet `tocloft`.

Ce paquet permet également de créer ses propres tables de contenu telle une table des théorèmes, une table des exemples... Dans ce document de cours, la table des exemples de code a été obtenue avec des fonctionnalités du paquet `floatrow` car elle traite de contenus flottants (voir page 48).

1. Dans le cas d'un article, les sections remplaçant les chapitres, la commande `\dosecttoc` remplace `\dominitoc` et la commande `\secttoc` remplace `\minitoc`.

2. Il peut arriver que des mini-tables des matières apparaissent décalées d'un ou plusieurs chapitres, le paquet `minitoc` pouvant être perturbé par des chapitres « étoilés ». Dans ce cas, la commande `\mtcaddchapter` (et `\mtcaddsection`) peut compenser un décalage d'un chapitre.

5.3 Les références

5.3.1 Les commandes usuelles

Pour faire référence à d'autres parties du document, trois commandes sont disponibles. La première, `\label{étiquette}`, se place à l'endroit où se trouve le passage à identifier. Ceci permet à \LaTeX de constituer une nouvelle entrée appelée *étiquette* dans sa liste interne de références. À chaque entrée est associée la page où elle se trouve et son numéro de division (ou de table, figure ou formule si elle est placée dans de tels éléments). Cette commande n'affiche donc rien.

Les deux autres commandes, `\ref{étiquette}` et `\pageref{étiquette}`, servent à afficher respectivement le numéro de la division (ou table, figure ou formule) et la page de la référence de l'étiquette indiquée. Une double compilation est nécessaire, sous peine d'observer un ?? ainsi qu'une erreur non bloquante dans le fichier `log`.

Un exemple apparaît en page 6... mais aussi ici.

5.3.2 Pour aller plus loin

Certains paquets permettent d'obtenir des systèmes de référence plus poussés. En voici deux :

- `varioref`, qui donne un système de référence plus naturelle avec l'utilisation d'expressions telle que « page suivante » ;
- `smartref`, qui permet d'accéder à plus d'informations lorsqu'une référence est faite, tel le numéro du chapitre seul (si on référence par exemple une sous-section) ou le nom du chapitre, de la section et ainsi de suite.

5.4 L'index

Un index, tel celui présent en page 111, s'obtient à l'aide de quatre commandes :

- la première est l'appel du paquet de gestion de l'index `makeidx`, dans le préambule.
- la deuxième commande, `\makeindex`, se place juste avant `\begin{document}`. Elle indique à \LaTeX de regrouper en un seul fichier tous les éléments à indexer. Ce fichier portera l'extension `.idx`.
- la troisième commande sert justement à désigner les éléments à indexer ainsi que les pages concernées. Sa version la plus simple est `\index{mot apparaissant dans l'index}`.
- enfin, la dernière commande, `\printindex`, permet d'indiquer dans le corps de document où sera disposé l'index.

```
\usepackage{makeidx}
\makeindex
\begin{document}
M. Torquemada\index{Torquemada}, inquisiteur, a été mis à l'index.
\printindex
C'est vraiment pas de chance pour M. Torquemada\index{Torquemada}.
\end{document}
```

CODE 5.4 – Disposition des commandes d'index

5.4.1 Le programme externe makeindex

En cas de compilation, \LaTeX n'affichera par défaut rien dans l'index. En effet, \LaTeX attend un élément bien précis pour générer l'index : les références triées selon un ordre alphabétique et regroupées entre elles. Ainsi, une référence citée sur trois pages différentes ne doit pas donner trois entrées d'index mais une entrée unique avec les trois numéros de page associés.

»Réfs
[13]40–42,228–235
[16]68–80
[31]55–56

»Réfs
[13]189–201
[16]659–694
[31]177–185

Cette liste organisée s'obtient en utilisant le programme `makeindex`³.

La première compilation crée le fichier `.idx` portant le même nom que le fichier `.tex`. Le programme `makeindex` doit traiter ce fichier⁴.

Est alors créé un fichier `.ind` qui contient l'ensemble trié des éléments à indexer. Une nouvelle compilation avec \LaTeX permet enfin d'intégrer ce fichier attendu par \LaTeX au document principal : l'index apparaît.

5.4.2 Quelques compléments sur l'indexation

La commande `\index{}` permet d'obtenir des entrées d'index assez détaillées. Voici les principales syntaxes disponibles — l'index de ce document offrant à chaque fois une illustration des effets générés — et combinables à merci :

- `\index{entrée !sous-entrée}` permet de créer une *entrée* présentant une *sous-entrée*. L'exemple est donnée par « compilation » présentant une sous-entrée « erreurs ». Une sous-sous-entrée peut être associée à la sous-entrée : il suffit d'ajouter un « ! » suivi de la sous-sous-entrée à la commande précédente.
- `\index{entrée|}` suivie plus loin de `\index{entrée|})` permet de donner une plage de pages associée à l'entrée au lieu de donner une liste de pages. L'exemple en est « structures ».
- `\index{entrée fictive@entrée}` permet de classer une *entrée* de l'index comme si elle devait occuper la place de l'entrée *fictive*. C'est le cas par exemple de l'entrée `\TeX` qui, comme elle correspond en fait à une commande commençant par `\`, est placée par défaut en début d'index⁵. Par le biais de cette commande, cette entrée est reclassée comme si elle valait le mot « tex ».

5.4.3 Pour aller plus loin

L'index peut être doté d'un style dédié. Ceci demande d'utiliser un fichier de style `.ist` et de modifier la commande d'appel. Certaines configurations peuvent également être obtenues par le biais d'un code \LaTeX . Ces différents cas sont présentés sur deux exemples à partir de la page 67.

La mécanique d'indexation présentée ici permet également de constituer des éléments comme un glossaire, une liste d'acronyme. Ceci implique l'utilisation de `glossaries`.

Enfin, il est possible de générer plusieurs index dans un même document avec le paquet `index`, comme le fait ce document.

Ces différentes utilisations avancées demandent de modifier les appels effectués par Texmaker. Par exemple, la compilation du glossaire demande d'exécuter une commande plus fine :

```
makeindex -s fichier.ist -t fichier.glg -o fichier.gls fichier.glo}
```

CODE 5.5 – Appel générique de `makeindex`

Dans cet appel de `makeindex`, le fichier suivant le `-s` est le fichier de style, celui suivant le `-o` est le nom du fichier résultat et celui suivant le `-t` est celui contenant le journal du traitement effectué.

5.5 La bibliographie

La composition d'une bibliographie constitue un élément important dans la création d'un document technique. Bien tenue, la bibliographie indique en général un souci de qualité de la part

3. Une autre possibilité revient à utiliser le programme `texindy` qui gère efficacement le tri des lettres accentuées mais qui demande une installation plus complète que celle fournie par défaut avec \LaTeX .

4. Dans le cadre de ce cours, cela revient à sélectionner le programme `makeindex` dans Texmaker et l'exécuter. Dans le cas de `texindy`, cela revient à lancer une ligne de commande « `texindy -F french fichier` ».

5. Les symboles sont classés avant les lettres dans l'ordre de l'index.

»Réfs
[13]201–227
[16]695–
770,771–826
[31]167–175

de son auteur. De ce point de vue, \LaTeX facilite largement cette tâche en la standardisant, nous évitant ainsi de nombreuses erreurs de présentation.

5.5.1 Le fichier bibliographique

Les données bibliographiques sont à regrouper sous une forme particulière dans un fichier à part. Ce principe a pour origine l'utilisation de \LaTeX pour traiter avec de véritables bases de données bibliographiques. L'extension de ce fichier sera `.bib`. Dans l'exemple qui suit est présenté le référencement minimum pour un livre (avec ici, en italique, un exemple).

```
@BOOK{identifiant,
  author = {Knuth, Donald E.},
  title = {The \TeX book},
  year = {1986},
  publisher = {Addison-Wesley}}
```

CODE 5.6 – Définition minimale d'un livre pour $\text{BIB}\TeX$

L'*identifiant* est un mot unique associé à l'ouvrage : il permet de faire appel à la référence documentaire de l'ouvrage dans la bibliographie.

Pour l'écriture du nom de l'auteur, la convention est d'écrire d'abord le nom, une virgule puis les prénoms. En cas d'auteurs multiples, ils s'écrivent les uns à suite des autres en étant séparés par un `and`. Enfin, pour éviter que $\text{BIB}\TeX$ ne fasse une erreur sur un nom de société ou d'organisme (en croyant lire un prénom suivi d'un nom), il est préférable de mettre entre accolades ce type de nom.

```
author = {Tic,A. and Tac,B. and Tic-Tic Tock,C.D. and {Tonck Corp.}},
```

CODE 5.7 – Cas d'auteurs multiples pour $\text{BIB}\TeX$

5.5.2 Le code \LaTeX

Dans le fichier `.tex`, à l'endroit où devra être disposée la bibliographie, se placent les trois lignes suivantes :

```
\bibliographystyle{style}
\bibliography{fichier}
\addcontentsline{toc}{chapter}{Bibliographie}
```

CODE 5.8 – Commandes pour la bibliographie

La première commande donne le *style* de présentation de la bibliographie ; ce style peut être par exemple `plain`, `alpha`, `apalike`, `abbrv`⁶. La deuxième commande indique le nom du *fichier* `.bib` qui contient nos références bibliographiques ; l'extension `.bib` ne doit pas être indiquée. La dernière commande, facultative, complète la table des matières selon la méthode vue en page 32.

5.5.3 Le contenu final

Ces deux étapes effectuées, il faut inclure dans le texte des liens vers les références bibliographiques comme cela est fait en fin de cette partie afin que \LaTeX les affiche bien dans la bibliographie.

6. Dans le cadre du document de cours est utilisé un style nommé `plain-fr` qui convertit en français le style `plain`. Ce style fait partie d'un ensemble nommé « fr-bib ». N'étant pas dans l'installation du cours, son fichier `plain-fr.bst` doit être placé dans le même répertoire que le fichier `.tex`.

phie. Pour effectuer cette référence, il faut utiliser la commande `\cite{identifiant}` où l'identifiant est celui figurant dans le fichier `◦bib`.

La commande `\nocite{*}` permet d'afficher toutes les références du fichier `◦bib`.

5.5.4 BIBTEX

Afin de générer la bibliographie définitive, il faut compiler le fichier `◦tex` avec un autre programme⁷ : `BIBTEX`. Ce dernier présente et trie les différentes références nécessaires tirées du fichier `◦bib`. Les éléments sont regroupés dans un fichier `◦bbl` qui sera pris en compte à la compilation suivante par `LTEX`.

Lors d'un changement de style de bibliographie, il est recommandé de supprimer le fichier `◦bbl` car ce dernier peut générer des erreurs entre styles bibliographiques.

5.5.5 Pour aller plus loin

Pour les bibliographies, un paquet se distingue de plus en plus des autres : `biblatex`. Ce dernier présente un ensemble de fonctionnalités⁸ répondant à de très nombreux besoins :

- suppression des fichiers de style. Le style et les tris sont en effet directement définis dans les options de la commande d'appel du paquet ;
- possibilité de gérer le style des citations d'ouvrage de la bibliographie dans le texte ;
- possibilité de générer des bibliographies par chapitre ou section ;
- possibilité de générer des bibliographies sectionnées par style d'ouvrage.

5.6 Les en-têtes et pieds de page

5.6.1 Les commandes usuelles

Quelques présentations courantes de ces zones sont proposées par défaut par `LTEX`. Ces présentations sont appelées par la commande `\pagestyle{style}` où le `style` peut être : `empty`, `plain`, `headings` ou `myheadings`.

Le style des pages est normalement commun à toutes les pages du document. Pour donner un style différent à une page spécifique, il existe la commande `\thispagestyle{style}`, à placer dans le code de la page concernée.

5.6.2 Pour aller plus loin

L'utilisation du paquet `fancyhdr` permet d'aller beaucoup plus loin dans le paramétrage des en-têtes et pieds de page, comme le montre l'exemple en 8.3.3, page 66.

5.7 Le titre

5.7.1 Les commandes usuelles

La commande `\maketitle` se place à l'endroit où se situera le titre ou la page titre. Cette commande va en fait rechercher quelques éléments placés dans le préambule du document :

- `\title{le titre}` ;
- `\author{l'auteur}` ;
- `\date{la date}`. Si cette commande n'est pas utilisée, `LTEX` affichera par défaut la date du jour de la compilation. Pour ne pas l'afficher, il suffit d'indiquer `\date{}`.

7. Le cours ne donne ici qu'un parmi quelques autres.

8. En français, la page de Bertrand MASSON [25] <<http://bertrandmasson.free.fr/index.php?article27/>>, tout comme la page de Franck PASTOR [28] <<http://www.cuk.ch/articles/7109>> donnent une bonne vision du sujet.

»Réfs
[13] 78–81, 508–512
[16] 219–236
[31] 43–44, 261–266

»Réfs
[13] 42–43
[16] —
[31] 41–42

- `\thanks{note}` qui s'utilise en lieu et place de `\footnote{note}` dans les trois commandes précédentes.

5.7.2 Pour aller plus loin

L'environnement `titlepage` permet également de gérer une page de titre personnalisée : cet environnement permet en effet de ne pas numéroté la page en cours (la numérotation commencera bien à 1 à la page suivante) et d'en retrancher les éventuels en-têtes et pieds de page. Le contenu est donc ici à créer. Ainsi, le paquet `pagedegarde` présenté en annexe D, page 102 utilise cet environnement.

Par ailleurs, des exemples de page (avec le code \LaTeX associé) regroupés par Peter WILSON peuvent être étudiés à l'adresse suivante : <http://www.ctan.org/pkg/titlepages>.

De même, un document de Yuri ROBBERS et Annemarie SKJOLD présente une méthode complète pour obtenir une couverture de livre (la première page et la dernière donc) à l'adresse suivante : <http://tug.org/pracjourn/2007-1/robbers/robbers.pdf>.

Chapitre 6

Les mathématiques

L^AT_EX est très réputé pour sa mise en page des mathématiques. Si le rédacteur doit bien faire attention au sens mathématique de ce qu'il écrit, le nombre de grands principes à suivre pour présenter des mathématiques reste assez limité.

6.1 Les modes mathématiques

6.1.1 Principes

Les mathématiques font appel à des règles de composition différentes de celles du texte classique : par exemple, les espacements horizontaux et verticaux sont gérés différemment, les variables sont composées en italique. L^AT_EX applique cet ensemble de règles lors qu'il est placé en **mode mathématique**, celui-ci se décomposant en deux variantes :

- le mode en ligne permet d'écrire des mathématiques directement dans des paragraphes de texte. Ce mode est obtenu en plaçant un symbole `$` de chaque côté du texte mathématique. Ce mode sert le plus souvent à évoquer les variables ou courts extraits de formules dans le texte.
- le mode séparé permet d'écrire une formule isolée du reste du texte : elle se retrouve centrée et séparée verticalement du texte. Pour obtenir ce mode, il faut introduire la formule par `\[` et la conclure par `\]`. L'environnement `equation` permet également de l'obtenir et ajoute par défaut une numérotation de l'équation.

```
Soit deux réels  $x$  et  $y$  tels que  $x > 2$  et  $y > 2$  alors  $x + y > 4$   
\begin{equation}  
x + y < x.y  
\end{equation}
```

Soit deux réels x et y tels que $x > 2$ et $y > 2$ alors

$$\begin{aligned}x + y &> 4 \\ x + y &< x.y\end{aligned}\tag{6.1}$$

CODE 6.1 – Modes mathématiques

6.1.2 Différents environnements pour équation

Il existe plusieurs environnements similaires à `equation` pour présenter différentes équations.

Style	Environnement
Une équation sans retour à la ligne	<code>equation</code>
Plusieurs équations à la suite	<code>gather</code>
Une équation sur plusieurs lignes	<code>multline</code>
Plusieurs équations alignées	<code>align</code>

TABLE 6.1 – Différents styles de présentation d'équation

Dans ces différents environnements, pour passer d'une ligne à une autre, il faut utiliser la commande `\\`. Dans le cas particulier de `align`, l'alignement vertical des équations se fait par rapport à des marqueurs indiqués par le caractère `&`. Dans l'exemple ci-dessous, l'alignement se fera visuellement sur le signe `=`.

```
Ceci conduit aux équations suivantes :
\begin{gather}
f(x) = (1-x)(1+z) \\
g(x) = 14-2x-z+xz \\
h(x) = f(x)+g(x) \quad \backslash\text{label}{eq1} \\
\end{gather}
Ce qui permet de déduire une expression simple de  $h$  en repartant de \ref{eq1} :
\begin{align}
h(x) &= (1-x)(1+z)+14-2x-z+xz & \backslash\text{nonnumber} & \\
h(x) &= 1-x+z-xz+14-2x-z+xz & \backslash\text{nonnumber} & \\
h(x) &= 15 - 3x \\
\end{align}
```

Ceci conduit aux équations suivantes :

$$f(x) = (1-x)(1+z) \quad (6.2)$$

$$g(x) = 14 - 2x - z + xz \quad (6.3)$$

$$h(x) = f(x) + g(x) \quad (6.4)$$

Ce qui permet de déduire une expression simple de h en repartant de 6.4 :

$$\begin{aligned} h(x) &= (1-x)(1+z) + 14 - 2x - z + xz \\ h(x) &= 1 - x + z - xz + 14 - 2x - z + xz \\ h(x) &= 15 - 3x \end{aligned} \quad (6.5)$$

CODE 6.2 – Equations et alignements

L'espacement du code est une fois encore libre, ce qui permet de mieux faire ressortir la structure souhaitée, comme l'illustre cet exemple.

6.1.3 Numérotation des équations

Ces différents environnements mathématiques disposent tous d'une version étoilée (comme par exemple `gather*`) qui supprime la numérotation des équations. Il est également possible de supprimer la numérotation d'une équation en plaçant les commandes `\notag` ou `\nonumber` sur la ligne ou les lignes où la numérotation doit être retranchée, comme cela est fait dans l'exemple précédent.

Les numéros d'équation peuvent être réutilisés avec `\label`, `\ref` et `\pageref`, comme vu en page 33. La commande `\label` doit être placée sur la ligne de l'équation numérotée souhaitée.

6.2 Les symboles mathématiques

Les symboles mathématiques s'obtiennent le plus souvent par des commandes qui ne fonctionnent qu'en mode mathématique. Sous réserve de charger les paquets `mathtools` et `amssymb`, \LaTeX dispose d'un très grand nombre de ces commandes pour lequel il existe un document de référence exhaustif[27]¹. Ce document a permis par exemple de trouver les symboles `\llbracket` et `\rrbracket` obtenus avec le paquet `stmaryrd` et utilisés dans la sous-section 6.4. Les listes présentées ici n'ont vocation qu'à donner une bonne base.

6.2.1 Opérateurs

\LaTeX rédige les mathématiques en italique. Traditionnellement, de nombreux opérateurs sont cependant rédigés en caractères romains ; aussi, quelques commandes donnent accès à cette présentation. En complément, certaines de ces commandes autorisent des mises en forme spécifiques (voir page 42).

Symbole	Commande	Symbole	Commande	Symbole	Commande
\cos	<code>\cos</code>	\sin	<code>\sin</code>	\tan	<code>\tan</code>
\arccos	<code>\arccos</code>	\arcsin	<code>\arcsin</code>	\arctan	<code>\arctan</code>
\cosh	<code>\cosh</code>	\sinh	<code>\sinh</code>	\tanh	<code>\tanh</code>
\exp ou e	<code>\exp</code> ou <code>e</code>	\min	<code>\min</code>	\cot	<code>\cot</code>
\ln	<code>\ln</code>	\max	<code>\max</code>	\coth	<code>\coth</code>
\log	<code>\log</code>	\inf	<code>\inf</code>	\lim	<code>\lim</code>
\lg	<code>\lg</code>	\sup	<code>\sup</code>	\liminf	<code>\liminf</code>
\Pr	<code>\Pr</code>	\det	<code>\det</code>	\limsup	<code>\limsup</code>

TABLE 6.2 – Opérateurs

Symbole	Commande	Symbole	Commande	Symbole	Commande
\leq	<code>\leq</code>	\geq	<code>\geq</code>	\neq	<code>\neq</code>
\leqslant	<code>\leqslant</code>	\geqslant	<code>\geqslant</code>	\sim	<code>\sim</code>
\ll	<code>\ll</code>	\gg	<code>\gg</code>	\simeq	<code>\simeq</code>
\lll	<code>\lll</code>	\ggg	<code>\ggg</code>	\approx	<code>\approx</code>
\lesssim	<code>\lesssim</code>	\gtrsim	<code>\gtrsim</code>	\equiv	<code>\equiv</code>

TABLE 6.3 – Égalités et inégalités

Symbole	Commande	Symbole	Commande	Symbole	Commande
\forall	<code>\forall</code>	\aleph	<code>\aleph</code>	\cup	<code>\cup</code>
\exists	<code>\exists</code>	∞	<code>\infty</code>	\cap	<code>\cap</code>
\nexists	<code>\nexists</code>	\emptyset	<code>\emptyset</code>	\subset	<code>\subset</code>
\in	<code>\in</code>	\varnothing	<code>\varnothing</code>	\supset	<code>\supset</code>
\notin	<code>\notin</code>	\setminus	<code>\setminus</code>	\vee	<code>\vee</code>
\neg	<code>\neg</code>	\circ	<code>\circ</code>	\wedge	<code>\wedge</code>
∇	<code>\nabla</code>	∂	<code>\partial</code>	\times	<code>\times</code>

TABLE 6.4 – Notations ensemblistes et fonctionnelles

1. Une autre source, plus originale, existe : `detexify`². Dessiner un symbole permet de trouver son code \LaTeX .

6.2.2 Lettres

Les lettres grecques s'obtiennent par des commandes portant leur nom : `\alpha` donnera ainsi α . Les majuscules² s'obtiennent en mettant une majuscule en première lettre : `\Omega` donnera donc Ω . Par ailleurs, quelques variantes graphiques existent pour certains caractères. La table suivante présente l'ensemble de ces lettres.

Symbole	Commande	Symbole	Commande	Symbole	Commande
α	<code>\alpha</code>	μ	<code>\mu</code>	ψ	<code>\psi</code>
β	<code>\beta</code>	ν	<code>\nu</code>	ω	<code>\omega</code>
γ	<code>\gamma</code>	ξ	<code>\xi</code>	\digamma	<code>\digamma</code>
δ	<code>\delta</code>	π	<code>\pi</code>	Γ	<code>\Gamma</code>
ϵ	<code>\epsilon</code>	ϖ	<code>\varpi</code>	Δ	<code>\Delta</code>
ε	<code>\varepsilon</code>	ρ	<code>\rho</code>	Θ	<code>\Theta</code>
ζ	<code>\zeta</code>	ϱ	<code>\varrho</code>	Λ	<code>\Lambda</code>
η	<code>\eta</code>	σ	<code>\sigma</code>	Ξ	<code>\Xi</code>
θ	<code>\theta</code>	ς	<code>\varsigma</code>	Π	<code>\Pi</code>
ϑ	<code>\vartheta</code>	τ	<code>\tau</code>	Σ	<code>\Sigma</code>
ι	<code>\iota</code>	υ	<code>\upsilon</code>	Υ	<code>\Upsilon</code>
κ	<code>\kappa</code>	ϕ	<code>\phi</code>	Φ	<code>\Phi</code>
\varkappa	<code>\varkappa</code>	φ	<code>\varphi</code>	Ψ	<code>\Psi</code>
λ	<code>\lambda</code>	χ	<code>\chi</code>	Ω	<code>\Omega</code>

TABLE 6.5 – Lettres grecques

LaTeX propose également quelques polices de caractères dédiées spécifiquement aux mathématiques :

- `\mathbb{majuscule}`. Ainsi, `\mathbb{N}` donne \mathbb{N} .
- `\mathcal{majuscule}`. Ici, `\mathbb{L}` donne \mathcal{L} .
- `\mathfrak{mot}` compose le mot en style gothique, tel `\mathfrak{Maths}` donnant $\mathfrak{M}\mathfrak{a}\mathfrak{t}\mathfrak{h}\mathfrak{s}$.

6.2.3 Flèches

Les symboles de flèches sont très nombreux mais peuvent se retrouver assez facilement sur la base des règles suivantes :

- la grande majorité des noms de ces symboles contient le terme `arrow`. Des exceptions courantes sont citées dans le tableau ;
- le sens de la flèche est donné par le terme `left` ou `right`. Le terme `leftright` donne une flèche à double sens ;
- le préfixe `long` donne une flèche plus longue ;
- une majuscule en début de nom donne un trait doublé ;
- les noms pour les flèches parallèles se finissent par `s`.

Symbole	Commande	Symbole	Commande	Symbole	Commande
\rightarrow	<code>\to</code>	\rightarrow	<code>\rightarrow</code>	\Leftrightarrow	<code>\Leftrightarrow</code>
\mapsto	<code>\mapsto</code>	\longrightarrow	<code>\longrightarrow</code>	\dashrightarrow	<code>\dashrightarrow</code>
\longmapsto	<code>\longmapsto</code>	\Rightarrow	<code>\Rightarrow</code>	\circlearrowright	<code>\circlearrowright</code>
\Leftrightarrow	<code>\iff</code>	\leftrightarrow	<code>\leftrightarrow</code>	\curvearrowright	<code>\curvearrowright</code>

TABLE 6.6 – Flèches

2. Tout du moins celles qui sont différentes de nos lettres majuscules classiques. En effet, l'alpha majuscule s'écrit par exemple à l'identique de notre A majuscule.

6.2.4 Accents

Quelques commandes permettent de compléter les symboles usuels avec des notations mathématiques courantes :

Symbole	Commande	Symbole	Commande	Symbole	Commande
\dot{a}	<code>\dot{a}</code>	\vec{a}	<code>\vec{a}</code>	\overline{abcd}	<code>\overline{abcd}</code>
\ddot{a}	<code>\ddot{a}</code>	\bar{a}	<code>\bar{a}</code>	\underline{abcd}	<code>\underline{abcd}</code>
\ddot{a}	<code>\dddot{a}</code>	\mathring{a}	<code>\mathring{a}</code>	\widehat{abcd}	<code>\widehat{abcd}</code>

TABLE 6.7 – Flèches

La commande `\vec` peut être remplacée par `\overrightarrow` pour des termes longs. Cette commande connaît d'ailleurs des variantes en remplaçant `over` par `under` ou `right` par `left`.

6.2.5 Notations graphiques

Quelques notations mathématiques se présentent sous forme géométrique. Elles se retrouvent souvent retenues pour d'autres usages, de la présentation le plus souvent. Comme vu précédemment, les commandes présentant le terme `left` disposent du symétrique `right`.

Symbole	Commande	Symbole	Commande	Symbole	Commande
•	<code>\bullet</code>	▲	<code>\blacktriangle</code>	△	<code>\triangle</code>
○	<code>\circ</code>	△	<code>\vartriangle</code>	△	<code>\bigtriangleup</code>
·	<code>\cdot</code>	◀	<code>\blacktriangleleft</code>	▽	<code>\bigtriangledown</code>
◯	<code>\bigcirc</code>	◁	<code>\vartriangleleft</code>	◁	<code>\triangleleft</code>
■	<code>\blacksquare</code>	▼	<code>\blacktriangledown</code>	♡	<code>\heartsuit</code>
□	<code>\square</code>	▽	<code>\triangledown</code>	◇	<code>\diamondsuit</code>
◆	<code>\blacklozenge</code>	★	<code>\bigstar</code>	♣	<code>\clubsuit</code>
◇	<code>\lozenge</code>	★	<code>\star</code>	♠	<code>\spadesuit</code>

TABLE 6.8 – Notations graphiques

6.3 Les exposants et indices

La mise en indice s'obtient par l'utilisation du caractère « `_` ». Ainsi, `2_n` donnera 2_n . Dès que l'indice contient plus d'un caractère, il faut l'encadrer avec des accolades pour définir l'indice comme un groupe. La mise en exposant se fait de manière identique avec le caractère « `^` ». Et ces deux règles peuvent se combiner.

Cette mise en indice et en exposant peut donner d'autres résultats lorsqu'elle est appliquée sur certaines commandes. Il en va ainsi sur les commandes comme `\lim`, de même que sur des symboles dits « grands opérateurs ».

Symbole	Commande	Symbole	Commande	Symbole	Commande
\sum	<code>\sum</code>	\int	<code>\int</code>	\bigcup	<code>\bigcup</code>
\prod	<code>\prod</code>	\iint	<code>\iint</code>	\bigcap	<code>\bigcap</code>
\coprod	<code>\coprod</code>	\iiint	<code>\iiint</code>	\bigwedge	<code>\bigwedge</code>
\oint	<code>\oint</code>	\iiint	<code>\iiint</code>	\bigvee	<code>\bigvee</code>

TABLE 6.9 – Grands opérateurs

Pour ces symboles et fonctions, en mode en ligne, les éléments mis en indice et exposant sont placés après le symbole. En mode séparé, les éléments se retrouvent au-dessous et au-dessus

du symbole. Le comportement obtenu en mode séparé peut être forcé en insérant la commande `\limits` entre le symbole et les éléments mis en indice et en exposant. L'inverse s'obtient avec la commande `\nolimits`.

```
Il convient de distinguer $ x_{yz} \neq x_{\{yz\}} $ et $ x^{yz} \neq x^{\{yz\}} $. De plus
l'expression $ A_{i_n} = \sum_{k=1}^n a_i^k $ est visuellement différente de
$ A_{i_n} = \sum\limits_{k=1}^n a_i^k $ mais cette dernière change la taille de
l'interligne, ce qui peut être assez peu esthétique au sein d'un paragraphe ayant un
interligne par ailleurs régulier.
\[ A_{i} = \lim_{n \rightarrow +\infty} \sum_{k=1}^n a_i^k \]
```

Il convient de distinguer $x_{yz} \neq x_{\{yz\}}$ et $x^{yz} \neq x^{\{yz\}}$. De plus l'expression $A_{i_n} = \sum_{k=1}^n a_i^k$ est visuellement différente de $A_{i_n} = \sum\limits_{k=1}^n a_i^k$ mais cette dernière change la taille de l'interligne, ce qui peut être assez peu esthétique au sein d'un paragraphe ayant un interligne par ailleurs régulier.

$$A_i = \lim_{n \rightarrow +\infty} \sum_{k=1}^n a_i^k$$

CODE 6.3 – Mises en indice et en exposant

6.4 Les délimiteurs

Les signes d'accolades, de crochets, de parenthèses et quelques autres moins courants cités ci-dessous peuvent être librement redimensionnés afin de rendre plus lisibles les équations.

Symboles	Commande	Symboles	Commande	Symboles	Commande
()	()	[]	\lceil \rceil		\lvert \rvert
[]	[]	[]	\lfloor \rfloor		\lVert \rVert
{ }	\{ \}	< >	\angle \rangle		

TABLE 6.10 – Délimiteurs

Pour cela, il suffit de faire précéder le signe à redimensionner de l'une des quatre commandes suivantes, par ordre croissant de taille : `\big`, `\bigg`, `\Big`, `\Bigg`.

```
Soit $\mathfrak{F}$ un flux financier tel que :
\[ \mathfrak{F} = \Big\{ \big(F_j, t_j\big) \Big| j \in \llbracket 0..n \rrbracket,
0 \leqslant t_1 < t_2 \dots < t_n, \quad \forall j \quad F_j \geqslant 0 \Big\}
```

Soit \mathfrak{F} un flux financier tel que :

$$\mathfrak{F} = \left\{ (F_j, t_j) \middle| j \in \llbracket 0..n \rrbracket, 0 \leqslant t_1 < t_2 \cdots < t_n, \quad \forall j \quad F_j \geqslant 0 \right\}$$

CODE 6.4 – Délimiteurs redimensionnés

Un ajustement de la taille de ces couples de symboles peut être fait automatiquement par \LaTeX , ce qui est tout particulièrement utile dans le cas des matrices. Il faut utiliser alors deux commandes : `\left` et `\right` suivies des symboles à redimensionner automatiquement. Ces deux

commandes ne peuvent qu'intervenir en couple. Pour le cas où il faudrait redimensionner un unique symbole, il faut que son symbole jumeau soit alors un point, ce dernier valant ici pour un symbole vide.

6.5 Racines et fractions

Les racines s'obtiennent avec `\sqrt{texte}` et les racines d'ordre `n` en ajoutant l'argument facultatif `[n]`.

Les fractions s'obtiennent avec `\frac{numérateur}{dénominateur}`.

Tracer le tableau de variation de la fonction f telle que :

$$f(x) = \sqrt[3]{\frac{1-x^2}{\sqrt{2}(1-x)^2}}$$

Tracer le tableau de variation de la fonction f telle que :

$$f(x) = \sqrt[3]{\frac{1-x^2}{\sqrt{2}(1-x)^2}}$$

CODE 6.5 – Racines et fractions

6.6 Les textes et espaces

Avec \LaTeX , le texte mathématique est composé en italique, sans accents. Pour pouvoir insérer du texte classique dans une formule en mode séparé, il convient d'utiliser la commande `\mathrm{texte}` : elle composera le texte en romain sans tenir compte des accents et espaces. Il existe également `\mathit{texte}`, `\mathbf{texte}`, `\mathtt{texte}` et `\mathsf{texte}` à l'image de ce qui a été vu dans la table 4.1 en page 16. Cependant, pour avoir espaces et accents, seule la commande `\text{texte}` est utilisable. Un exemple est donné dans le code 6.7, page 45.

L'espacement peut également être modifié à l'aide des quelques commandes suivantes.

Espace	Commande
■ ■	<code>\,</code>
■ ■	<code>\;</code>
■ ■	<code>\enskip</code>
■ ■	<code>\quad</code>
■ ■	<code>\qquad</code>

TABLE 6.11 – Espacements mathématiques

6.7 Les matrices

Quelques environnements permettent de présenter des matrices. Ces environnements, à la différence des environnements comme `equation` vus précédemment, ne génèrent pas le mode mathématique. Ils doivent donc être placés en mode mathématique par d'autres commandes usuelles. Par contre, les environnements de matrice partagent les mêmes principes d'écriture déjà rencontrés dans ce document :

- pour passer d'une cellule à une autre (sur une ligne) d'une matrice, il faut utiliser le caractère `&` ;
- pour passer d'une ligne à une autre, il faut utiliser `\\`.

Style	Environnement
Sans séparateurs	<code>matrix</code>
Avec parenthèses	<code>pmatrix</code>
Avec accolades	<code>Bmatrix</code>
Avec crochets	<code>bmatrix</code>
Avec filets verticaux	<code>vmatrix</code>
Avec doubles filets verticaux	<code>Vmatrix</code>

TABLE 6.12 – Différents styles de présentation de matrice

Il existe par ailleurs pour les matrices quelques symboles dédiés, indiqués dans l'exemple ³.

```
Le déterminant, noté  $\det$ , est défini pour une matrice comme : \begin{equation}
\det\left(\begin{matrix} a_{1;1} & \cdots & a_{1;n} \\ \vdots & \ddots & \vdots \\ a_{n;1} & \cdots & a_{n;n} \end{matrix}\right) = \begin{vmatrix} a_{1;1} & \cdots & a_{1;n} \\ \vdots & \ddots & \vdots \\ a_{n;1} & \cdots & a_{n;n} \end{vmatrix} \\
= \sum_{\sigma \in \mathfrak{S}_n} \varepsilon(\sigma) \prod_{i=1}^n a_{\sigma(i),i}
\end{equation}
```

Le déterminant, noté \det , est défini pour une matrice comme :

$$\det \left(\begin{bmatrix} a_{1;1} & \cdots & a_{1;n} \\ \vdots & \ddots & \vdots \\ a_{n;1} & \cdots & a_{n;n} \end{bmatrix} \right) = \begin{vmatrix} a_{1;1} & \cdots & a_{1;n} \\ \vdots & \ddots & \vdots \\ a_{n;1} & \cdots & a_{n;n} \end{vmatrix} = \sum_{\sigma \in \mathfrak{S}_n} \varepsilon(\sigma) \prod_{i=1}^n a_{\sigma(i),i} \quad (6.6)$$

CODE 6.6 – Exemple de matrice

Enfin et plus largement, les mathématiques peuvent aussi se servir de tableaux, comme présenté en page 47, pour obtenir certaines présentations.

```
La variable aléatoire  $X$  suit une loi de Bernoulli de paramètre  $p$  si et si seulement
elle vérifie les égalités suivantes :
\left[ \begin{array}{ccc}
X=0 & \text{avec la probabilité} & p \\
X=1 & \text{avec la probabilité} & (1-p)
\end{array} \right.
Dans ce cas, on notera  $X \sim \mathcal{B}e(p)$ .
```

La variable aléatoire X suit une loi de Bernoulli de paramètre p si et si seulement elle vérifie les égalités suivantes :

$$\begin{cases} X = 0 & \text{avec la probabilité} & p \\ X = 1 & \text{avec la probabilité} & (1-p) \end{cases}$$

Dans ce cas, on notera $X \sim \mathcal{B}e(p)$.

CODE 6.7 – Utilisation d'un tableau

3. Source : [Wikipédia](#), où il est possible de récupérer directement le code \LaTeX des mathématiques en affichant les propriétés des images présentant les équations.

6.8 Un élément d'actuariat

Un paquet, `lifecon`, permet d'utiliser des commutations actuarielles, tout particulièrement la notation des temporaires « *exemple* » avec la commande `\lcroof`. Ce paquet n'est pas présent sur les serveurs de paquets \LaTeX . Pour l'utiliser, il suffit de placer le fichier `lifecon.sty` dans le même répertoire que le fichier `.tex` que vous compilez⁴.

Un autre intérêt de ce paquet réside dans sa documentation⁵ qui précise comment obtenir la plupart des commutations actuarielles classiques.

En voici deux exemples appliqués :

```
La formule générale d'un capital différé de $n$ pour un groupe $G=(xyz\dots m)$ au
premier décès est :
\begin{align*}
{}_nE_{xyz\dots m} &= v^n \cdot {}_n p_{xyz\dots m} \\
&= v^n \cdot {}_n p_x \cdot {}_n p_y \cdots {}_n p_m
\end{align*}
La formule de l'annuité temporaire de $n$ années pour un groupe au dernier décès à
deux têtes est :
\[ \overline{a}_{xy:\overline{n}|} = \overline{a}_{x:\overline{n}|} + \overline{a}_{y:\overline{n}|} - \overline{a}_{xy:\overline{n}|} \]
```

La formule générale d'un capital différé de n pour un groupe $G = (xyz\dots m)$ au premier décès est :

$$\begin{aligned} {}_nE_{xyz\dots m} &= v^n {}_n p_{xyz\dots m} \\ &= v^n {}_n p_x {}_n p_y \cdots {}_n p_m \end{aligned}$$

La formule de l'annuité temporaire de n années pour un groupe au dernier décès à deux têtes est :

$$\overline{a}_{\overline{xy}:\overline{n}|} = \overline{a}_{x:\overline{n}|} + \overline{a}_{y:\overline{n}|} - \overline{a}_{xy:\overline{n}|}$$

CODE 6.8 – Formules de mathématiques des assurances

4. Des solutions automatisées mais elles demandent la bonne connaissance de la mécanique d'installation de \LaTeX .

5. Voir <http://maths.dur.ac.uk/stats/courses/AMII/LifeConSymbolsGuide.pdf>.

Chapitre 7

Les flottants : tableaux et figures

Un **flottant** représente un objet que \LaTeX place en respectant au mieux des règles de présentation de page, quitte à reporter l’affichage de l’objet sur une autre page, typiquement lorsque trop d’objets figurent sur la même page. La place de l’objet peut donc varier au fur et à mesure de l’évolution du document.

Il existe deux types de flottants classiques : les tableaux et les figures. D’autres peuvent être créés, tels les « codes » présents dans ce document.

7.1 Les tableaux

Les tableaux sont des objets typographiquement complexes : sur la plupart des logiciels de traitement de texte, ils demandent beaucoup de réglages dès que les exigences en terme de présentation sont élevées. \LaTeX ne fait pas exception à cette règle. Il faut en général décrire toute la structure des tableaux mais il existe quelques solutions pour gagner en concision.

»Réfs
[13]103–144
[16]243–288
[31]63–72

7.1.1 Les règles de base

Les tableaux s’obtiennent en général en utilisant l’environnement `tabular`. Cet environnement nécessite un paramètre spécifique : le « format-colonne »¹, autrement dit la présentation de chaque colonne. Le format de chaque colonne est représenté par une lettre dont la valeur indique la présentation :

- `l` pour un alignement à gauche ;
- `c` pour un centrage ;
- `r` pour un alignement à droite.

Dans le cas ci-après, la présentation centre la première colonne et aligne à gauche les trois autres colonnes avec `clll`.

Une fois le format-colonne établi, le contenu du tableau lui-même se présente en tenant compte des deux règles suivantes, tout à fait similaires à celles vues dans les matrices en page 44 :

- pour passer d’une cellule à une autre sur une ligne du tableau, il faut utiliser le caractère `&` ;
- pour passer d’une ligne à une autre, il faut utiliser `\\`.

1. Terminologie maison.

```

Du texte.
\begin{tabular}{c111}
Mois      & Effectif      & Moyenne      & \'{E}cart-type  & \\
Octobre   & 735           & 4,49\%       & 0,41\%         & \\
Novembre  & 756           & 4,36\%       & 0,37\%         & \\
D cembre  & 812           & 4,22\%       & 0,22\%         & \\
\end{tabular}
Et encore du texte.

```

	Mois	Effectif	Moyenne	�cart-type	
Du texte.	Octobre	735	4,49%	0,41%	Et encore du texte.
	Novembre	756	4,36%	0,37%	
	D�cembre	812	4,22%	0,22%	

CODE 7.1 – Tableau simple

Pour rappel, \LaTeX considère que les espaces multiples sont  quivalents   un espace unique. Ceci permet de pr senter au besoin un peu plus clairement les tableaux dans le code source de son document, comme dans cet exemple.

Par ailleurs, cet exemple montre qu'un tableau, s'il est plac  directement dans le cours du texte, s'ins re comme un bloc centr  verticalement par rapport   la ligne de texte.

7.1.2 Le passage en flottant

Cette pr sentation peut  tre am lior e en d finissant le tableau comme un objet flottant avec l'environnement `table`.

Il reste cependant possible d'indiquer   \LaTeX o  placer *de pr f rence* sur une page le flottant gr ce   quelques indications plac es entre crochets juste apr s `\begin{table}`, leur ordre indiquant l'ordre de priorit  dans la pr sentation. Les indications possibles sont :

- `h` : ici, autrement dit juste apr s le texte pr c dant le tableau dans le fichier `.tex` ;
- `t` : en haut de page ;
- `b` : en bas de page ;
- `p` : sur une page s par e ;
- `!` : force un peu le respect de la r gle indiqu e.

Ainsi, dans l'exemple ci-apr s, il est demand  d'afficher le tableau   l'endroit o  il est pr sent dans le code et, si cela n'est pas possible, en haut de page.

Ce passage en flottant permet par ailleurs de cr er de l'espace autour de ce tableau et d'ajouter une l gende avec la commande `\caption{l gende}`. La l gende sera num rot e et pourra  tre reprise dans une liste de tableaux g n r e par la commande `\listoftables`, proche voisine de `\tableofcontents`, tout comme le num ro pourra  tre repris dans les m caniques de r f rences vues en page 33.

```
Voici le tableau-\ref{tablo} :
\begin{table}[!ht]
\centering
\begin{tabular}{cccc}
Mois & & Effectif & Moyenne & & \'{E}cart-type & \\
Octobre & & 735 & 4,49\% & & 0,41\% & \\
Novembre & & 756 & 4,36\% & & 0,37\% & \\
D cembre & & 812 & 4,22\% & & 0,22\% & \\
\end{tabular}
\caption{Taux de r ussite} \label{tablo}
\end{table}
```

Voici le tableau 1.1 :

Mois	Effectif	Moyenne	�cart-type
Octobre	735	4,49%	0,41%
Novembre	756	4,36%	0,37%
D�cembre	812	4,22%	0,22%

TABLE 1.1 – Taux de r ussite

CODE 7.2 – Tableau flottant

Pour accompagner le passage en flottant, le tableau a  t  centr  avec l'utilisation de la commande `\centering`. Sans cela, le tableau para trait d centr  par rapport au texte de la l gende.

7.1.3 Les filets

Un tableau se pr sente souvent avec des filets, traits s parant lignes et colonnes. Ils s'obtiennent respectivement par la commande `\hline` et par l'ajout dans le format-colonne du symbole « | ». Avec cette m thode, les filets peuvent  tre doubl s en doublant la commande.

```
\begin{table}[!ht]
\centering
\begin{tabular}{|c|c|c|c|}
\hline
Mois & Effectif & Moyenne & \'{E}cart-type \\
Octobre & 735 & 4,49\% & 0,41\% \\
Novembre & 756 & 4,36\% & 0,37\% \\
D cembre & 812 & 4,22\% & 0,22\% \\
\end{tabular}
\caption{Taux de r ussite}
\end{table}
```

Mois	Effectif	Moyenne	�cart-type
Octobre	735	4,49%	0,41%
Novembre	756	4,36%	0,37%
D�cembre	812	4,22%	0,22%

TABLE 1.1 – Taux de r ussite

CODE 7.3 – Tableau avec filets

La commande `\cline` permet d'obtenir des filets partiels. Ainsi, `\cline{1-2}` indique ici qu'un filet horizontal va de la colonne 1   la colonne 2. Cette fonction sera particuli rement utile lors de la fusion de lignes, vue ci-apr s.

Le paquet `hhline` offre ici une alternative. La commande qu'il définit permet de déterminer de quelle manière le filet est tracé au niveau de chaque cellule. Le contenu de la commande est une suite de caractère, un par cellule, selon la codification suivante :

- `-` donne un filet simple ;
- `=` donne un filet double ;
- `~` n'affiche pas le filet.

7.1.4 La fusion de cellules

Des cellules de la même ligne peuvent être fusionnées grâce à la commande :

`\multicolumn{nb de colonnes}{présentation}{contenu}`

où *présentation* correspond aux valeurs `r`, `l` et `c` avec ou sans les barres verticales `|`.

De même, des cellules de la même colonne peuvent être fusionnées par la commande fournie par le paquet `multirow` :

`\multirow{nb de lignes}{largeur}{contenu}`

Ici, *largeur* correspond à la largeur des cellules fusionnées en colonne. Cette dimension peut être déterminée par \LaTeX en remplaçant `{largeur}` par `*`. Dans l'exemple du tableau 7.4, il peut être indiqué `{1cm}` ou `*` pour un effet presque similaire.

Cette commande introduit cependant une anomalie car `\hline` ne traite pas la fusion de cellule : le filet fait systématiquement toute la largeur du tableau. Il faut donc retenir `\cline{2-4}`.

```
\begin{table}[!ht]
\centering
\begin{tabular}{|c||c|c|c|c|}
\hline
\multirow{2}{*}{Mois} & \multicolumn{3}{c|}{Statistiques} & \\
& Effectif & Moyenne & & \'{E}cart-type \\
\hline
Octobre & 735 & 4,49\% & 0,41\% & \\
Novembre & 756 & 4,36\% & 0,37\% & \\
D cembre & 812 & 4,22\% & 0,22\% & \\
\end{tabular}
\caption{Taux de r ussite}
\end{table}
```

Mois	Statistiques		
	Effectif	Moyenne	�cart-type
Octobre	735	4,49%	0,41%
Novembre	756	4,36%	0,37%
D�cembre	812	4,22%	0,22%

TABLE 1.1 – Taux de r ussite

CODE 7.4 – Tableau avec fusion de cellules

7.1.5 L'enrichissement du format-colonne

Le paquet `array` permet d'approfondir la d finition du format-colonne du tableau. Il devient possible de d finir la largeur de chaque colonne avec la nouvelle balise `m{dimension}`.

De plus, il est possible d'introduire dans ce format-colonne des commandes   appliquer   chacune des cellules de chaque colonne par le biais des commandes `>{commandes}` et `<{commandes}`. La premi re place des commandes avant chaque cellule de la colonne, la seconde apr s chaque cellule de la colonne. Dans l'exemple qui suit, le recours   ces commandes permet d'obtenir une  criture pench e dans la premi re colonne et un centrage dans les trois autres colonnes, le centrage ne pouvant pas  tre obtenu par le `c` remplac  ici par `m{}`.

```

\begin{table}[!ht]
\centering
\begin{tabular}{|>{\slshape}c|m{2cm}<{\centering}|m{2cm}<{\centering}|
m{2cm}<{\centering}|}
\hline
Mois      & Effectif & Moyenne & \'{E}cart-type & \\ \hline
Octobre   & 735      & 4,49\%  & 0,41\%        & \\ \hline
Novembre  & 756      & 4,36\%  & 0,37\%        & \\ \hline
D cembre  & 812      & 4,22\%  & 0,22\%        & \\ \hline
\end{tabular}
\caption{Taux de r ussite}
\end{table}

```

Mois	Effectif	Moyenne	�cart-type
Octobre	735	4,49%	0,41%
Novembre	756	4,36%	0,37%
D�cembre	812	4,22%	0,22%

TABLE 1.1 – Taux de r ussite

CODE 7.5 – Tableau avec  galisation des largeurs de colonne

7.1.6 Un peu de couleur

Le paquet `colortbl`² met   disposition trois commandes permettant de colorer le fond des cellules d'un tableau :

- `\columncolor{couleur}`, dans le format-colonne, applique la *couleur*   la colonne.
- `\rowcolor{couleur}`, dans le tableau, applique la *couleur*   la ligne o  elle se situe.
- `\cellcolor{couleur}`, dans la cellule concern e, applique la *couleur*.

```

\begin{table}[!ht]
\centering
\begin{tabular}{|>{\columncolor{bleu8}\slshape}c|m{2cm}<{\centering}|
m{2cm}<{\centering}|m{2cm}<{\centering}|}
\hline \rowcolor{bleu7}
Mois      & Effectif & Moyenne & \'{E}cart-type & \\ \hline
Octobre   & 735      & \cellcolor{orange8}4,49\% & 0,41\%        & \\ \hline
Novembre  & 756      & 4,36\%  & 0,37\%        & \\ \hline
D cembre  & 812      & 4,22\%  & 0,22\%        & \\ \hline
\end{tabular}
\caption{Taux de r ussite}
\end{table}

```

Mois	Effectif	Moyenne	�cart-type
Octobre	735	4,49%	0,41%
Novembre	756	4,36%	0,37%
D�cembre	812	4,22%	0,22%

TABLE 1.1 – Taux de r ussite

CODE 7.6 – Tableau avec cellules color es

  ces effets sur le fond des cellules, peuvent s'ajouter deux autres bascules :

2. Ce paquet est activ  de pr f rence avec la commande `\usepackage[table]{xcolor}`. En effet, ce chargement r cup re `colortbl` mais aussi la commande `\rowcolors` vue par la suite.

- `\rowcolors{ligne}{couleur1}{couleur2}` qui se place avant le tableau et qui fait alterner la couleur de fond de chaque ligne entre *couleur1* (ligne impaire) et *couleur2* (ligne paire) à partir de la ligne de numéro *ligne* ;
- `\arrayrulecolor{couleur}` change la couleur des filets.

```
\begin{table}[!ht]
\centering
\arrayrulecolor{bleu5}
\rowcolors{2}{bleu9}{white}
\begin{tabular}{|>{\slshape}c||m{2cm}<{\centering}|m{2cm}<{\centering}|
m{2cm}<{\centering}|}
\hline \rowcolor{bleu8}
Mois & Effectif & Moyenne & & \'{E}cart-type & \\ \hline \hline
Octobre & 735 & & \cellcolor{orange8}4,49\% & 0,41\% & \\ \hline
Novembre & 756 & & 4,36\% & 0,37\% & \\ \hline
D cembre & 812 & & 4,22\% & 0,22\% & \\ \hline
\end{tabular}
\caption{Taux de r ussite}
\arrayrulecolor{black}
\end{table}
```

Mois	Effectif	Moyenne	�cart-type
Octobre	735	4,49%	0,41%
Novembre	756	4,36%	0,37%
D�cembre	812	4,22%	0,22%

TABLE 1.1 – Taux de r ussite

CODE 7.7 – Tableau avec cellules et filets color es

7.1.7 Une saisie simplifi e

Si la pr sentation d'un tableau reste souvent d licate, la saisie des  l ments s parant cellules et lignes (`&` et `\\`) peut  tre simplifi e. En g n ral, les logiciels pens s pour travailler du code \LaTeX proposent des  diteurs de tableaux simples g n rant le code une fois des cellules saisies. Ici, nous illustrons une autre solution utilisant Excel : le recours   la macro Excel2LaTeX³ qui convertit un tableau quelconque en un tableau \LaTeX .

Le code de cette commande est librement accessible et modifiable. Est fourni avec ce cours un exemple de modification du code permettant de g rer plus finement certains aspects : encadrement des cellules, traitements de certains caract res sp ciaux. Le r sultat, un code pas forc ment dig st  du fait de l'important nombre de commandes utilisant fortement les possibilit s de `\multicolumn`, donne un r sultat pr sent  ci-apr s.

3. Elle est t l chargeable   l'adresse <http://www.ctan.org/tex-archive/support/excel2latex>.

```

\begin{table}[!ht]
\centering
\begin{tabular}{|c|m{2cm}|m{2cm}|m{2cm}|}
\hline{----}
\multicolumn{1}{|E|}{\textbf{Mois}} &
\multicolumn{1}{|E|}{\textbf{Effectif}} &
\multicolumn{1}{|E|}{\textbf{Moyenne}} &
\multicolumn{1}{|E|}{\textbf{'{E}cart-type}}
\\
\hline{----}
\multicolumn{1}{|B|}{Octobre} &
% Et le code continue ainsi cellule après cellule jusqu'à clôture du tableau.

```

Mois	Effectif	Moyenne	Écart-type
Octobre	735	4,49%	0,41%
Novembre	756	4,36%	0,37%
Décembre	812	4,22%	0,22%

TABLE 1.1 – Taux de réussite

CODE 7.8 – Tableau issu d'une commande Excel dédiée

7.1.8 Pour aller plus loin

Dans le cas où les tableaux demandent plusieurs pages pour être affichés, le recours au paquet `longtable` est recommandé. Ce dernier demande une double compilation pour présenter proprement les tableaux.

Les filets peuvent également être modifiés, en particulier par le paquet `arydshln`.

7.2 Les images

L'insertion d'images s'obtient avec le paquet `graphicx`. Ce paquet présente l'option `draft` qui permet d'afficher le document sans y intégrer les images. Elles sont remplacées alors par des cadres indiquant l'espace qu'elles vont occuper.

➔Réfs
[13]393–453
[16]607–660
[31]123–
135,315–325

7.2.1 Le format et la localisation des images

Selon le type de compilation retenue (voir page 8), nous devons convertir nos images en format `eps` (chaîne `ps2pdf`) ou en format `jpg` (chaîne `pdflatex`) pour pouvoir les insérer⁴.

De plus, les images ne sont pas forcément dans le répertoire où se trouve le fichier `.tex`. Dans ce cas, il faut préciser le chemin pour les trouver en partant du dossier où est le fichier `.tex`. Indiquer comme nom de répertoire `..` indique que \LaTeX doit passer au dossier parent. Ainsi le chemin `../Test/Image.jpg` fait passer dans le dossier parent puis le dossier `Test` pour y trouver l'image `Image.jpg`.

7.2.2 L'insertion des images

Pour insérer les images, le paquet `graphicx` met à disposition la commande `\includegraphics`.

```

\includegraphics[options]{adresseimage}

```

CODE 7.9 – Insertion d'images

4. En effet, le passage systématique de nos documents en format postscript impose la seule reconnaissance de ce format d'image. Voir annexe B.1.

Les *options* courantes sont :

- `height` pour indiquer la hauteur de l'image dans une unité de mesure acceptée ;
- `width` pour indiquer de façon similaire la largeur de l'image ;
- `angle` pour indiquer l'angle de la rotation à appliquer à l'image (sans unité).
- `scale` pour indiquer un facteur d'échelle à l'image (sans unité).

Les exemples qui suivent sont présentés avec des options gérant la taille de l'image à l'affichage. Il est à noter que les images utilisées pour notre exemple ont été placées dans un sous-répertoire nommé `images`.

```
Voici une image qui arrive à l'improviste...
\includegraphics[width=2cm]{images/lettres.jpg}
et qui nous donne une présentation de texte quelque peu perturbée.
```



Voici une image qui arrive à l'improviste... et qui nous donne une présentation de texte quelque peu perturbée.

CODE 7.10 – Insertion d'une image dans du texte

7.2.3 Le passage en flottant

De la même manière que les tableaux avec l'environnement `table`, les figures peuvent être gérées comme des flottants avec l'environnement `figure`. Ceci permet d'obtenir la liste des figures avec `listoffigures`. En voici un exemple.

```
\begin{figure}[!ht]
\centering
\includegraphics[width=9cm]{images/lettres.jpg}
\caption{\'{E}nigme}
\end{figure}
```



FIGURE 1.1 – Énigme

CODE 7.11 – Insertion d'une image

L'image suivante voit sa hauteur définie et subit une rotation. Spécifier la hauteur et la largeur

de l'image conduit à une déformation de cette dernière pour qu'elle tienne dans les dimensions indiquées.

```
\begin{figure}[!ht]
\centering
\includegraphics[height=2.5cm,angle=30]{images/lettres.jpg}
\includegraphics[height=2.5cm,angle=-30]{images/lettres.jpg}
\caption{Seconde image}
\end{figure}
```



FIGURE 1.1 – Seconde image

CODE 7.12 – Insertion d'une image avec rotation

7.2.4 Commandes complémentaires

Le paquet `graphicx` propose quelques commandes complémentaires assez utiles pour repousser quelques limitations courantes de \LaTeX . Ces commandes ont pour point commun de traiter des boîtes et leur *contenu*, comme vu en page 22.

```
\reflectbox{contenu}
\scalebox{ratio}{contenu}
\resizebox{hauteur}{largeur}{contenu}
\rotatebox{angle}{contenu}
```

CODE 7.13 – Boîtes avec `graphicx`

Dans le cas de `\resizebox`, un redimensionnement proportionnel est possible en remplaçant la dimension à garder proportionnelle par un `!`. Pour le reste, les exemples ci-dessous illustrent les fonctions de ces boîtes.

```
Il est parfois \reflectbox{renversant} comme \scalebox{-1}{perturbant} de découvrir
\resizebox{3cm}{7pt}{l'étendue} de certaines fonctionnalités vraiment aux
\rotatebox{-30}{bords} des \scalebox{0.3}{limites}, \resizebox{5cm}{!}{n'est-ce-pas?}
```

Il est parfois *renversant* comme *perturbant* de découvrir l'*étendue* de cer-
taines fonctionnalités vraiment aux *bords* des *limites*, *n'est-ce-pas ?*

CODE 7.14 – Exemple de déformations de texte

7.2.5 Pour aller plus loin

Le paquet `picins` permet de disposer des images à côté du texte, comme c'est le cas avec la photographie présente en page 2. Pour simplifier sa présentation, il met à disposition la commande suivante⁵ :

```
\parpic[encadrement]{\includegraphics{adresseimage}}
```

CODE 7.15 – Commande d'insertion d'une image

Le paramètre *encadrement* donne la position par rapport au texte (`l` ou `r` selon que l'image soit positionnée à gauche ou à droite) ainsi que le format de l'éventuel cadre autour de l'image. Les possibilités sont les suivantes :

- `d` pour *dashed*, un cadre de tirets ;
- `f` pour *frame*, un cadre simple ;
- `o` pour *oval*, un cadre à coins arrondis ;
- `s` pour *shadow*, un cadre avec ombre ;
- `x` pour *box*, un cadre « fil de fer ».

Ce paquet permet de générer une *légende* à l'image de `\caption{légende}` vue précédemment avec la commande `\piccaption{légende}`. Cette commande est à placer juste avant l'image.

5. La commande s'avère un peu plus étoffée que cela, comme le précise la documentation de `picins`.

Troisième partie

Notions avancées

Chapitre 8

La programmation avec L^AT_EX

Les éléments présentés ci-après permettent de modifier ou d'affiner le comportement de L^AT_EX, d'automatiser certaines opérations et d'obtenir des fonctionnalités plus complexes. Il faut cependant garder en tête que les erreurs que peuvent déclencher ces fonctionnalités s'avèrent parfois un peu plus difficiles à analyser que les autres.

8.1 La scission du code

Un fichier `.tex` peut intégrer à tout moment le contenu d'autres fichiers `.tex`. Ceci rend plus lisible le code de chaque fichier L^AT_EX par découpage en fichiers de taille réduite¹. Ainsi dans le document source de ce cours, l'ensemble des données tracées dans la figure 9.1 sont placées dans un fichier `.tex` séparé. La commande associée s'appelle `\input{fichier}` où le nom du *fichier* est donné sans l'extension `.tex`.

La commande `\include{fichier}`, plus orientée sur la rédaction de longs documents, a le même effet que la commande précédente à ceci près :

- elle insère un saut de page si le fichier contenu ne le fait pas déjà (par exemple en commençant un nouveau chapitre) ;
- elle peut être désactivée par la commande `\includeonly{fichiers}`. Cette dernière liste les *fichiers* qui seront insérés, les noms des fichiers étant séparés par des virgules, sans indication des `.tex`. Les autres fichiers indiqués par des `\include` seront refusés. Ceci permet de contrôler en un unique endroit l'insertion ou pas de différents sous-documents.

L'intérêt de `\include` est de pouvoir compiler et visualiser différents morceaux choisis de son document sans pour autant compiler la totalité. Ceci fait gagner du temps pour la constitution de documents longs.

8.2 La définition de nouvelles commandes

8.2.1 Principes

Les commandes personnelles permettent de simplifier l'obtention de résultats complexes et d'uniformiser certaines actions, garantissant une bonne unité de traitement ou de présentation. Ces nouvelles commandes peuvent être définies à l'aide de la commande suivante :

1. L'utilisation des paquets s'inscrit totalement dans cette démarche. Ne pas faire cela aurait donné des préambules terriblement longs !

»Réfs
[13]259–260
[16]20–23
[31]56–57

»Réfs
[13]241–261
[16]855–903
[31]127–152

```
\newcommand{\nom}[nombre de paramètres]{définition}
```

CODE 8.1 – Nouvelle commande

Le *nom* de commande est composé de lettres non accentuées et doit être toujours précédé du `\` indiqué ci-dessus. Si le nom est déjà utilisé, \LaTeX génère l'erreur « *LaTeX Error : Command already defined.* ». Dans ces cas-là, sous réserve que la commande d'origine ne serve pas à d'autres besoins, il faut utiliser la commande `\renewcommand` qui a la même syntaxe que `\newcommand`.

La *définition* est une suite de commandes \LaTeX . Lors de l'écriture de cette définition, les différents paramètres sont appelés en utilisant la notation « `#x` », où `x` est le numéro du paramètre. Il peut y avoir jusqu'à neuf paramètres pour une commande.

Il est d'usage de placer les définitions et redéfinitions de commandes dans le préambule pour faciliter la relecture des documents mais elles peuvent être également placées dans le corps pour permettre de modifier le comportement de \LaTeX en cours de document.

Exemple sans paramètre

Avec ce système, il est possible de définir ses propres « constantes » avec des commandes sans paramètres. Ci-dessous, la première définit la commande `\InBT` pour laquelle la définition est un simple texte, texte qu'elle va donc écrire. La seconde, `\ba`, donne un nom de commande plus court pour appeler la contre-oblique.

```
\newcommand{\InBT}{inégalité de Bienaymé-Tchebyshev}
\newcommand{\ba}{\textbackslash}
Dans l'\InBT, pas de symbole \ba.
```

.....

Dans l'inégalité de Bienaymé-Tchebyshev, pas de symbole \.

CODE 8.2 – Définitions de commande sans argument

Exemple avec paramètres

Pour ce document, il a été défini la commande `\macro` avec un argument unique qui permet d'afficher en texte « machine à écrire » le nom de commande précédé du symbole de la contre-oblique. En voici la définition :

```
\newcommand{\macro}[1]{\texttt{\textbackslash #1}}
Voici un test de ma nouvelle commande : \macro{macrocommande}.
```

.....

Voici un test de ma nouvelle commande : \macrocommande.



CODE 8.3 – Définition de commande à un argument

La commande ci-après permet de générer quatre nuances d'une même couleur, les trois paramètres étant le nom de la couleur, le système de couleur et la composition de la couleur dans ce système. Chaque couleur ainsi créée a un nom finissant par un chiffre compris entre 0 et 3.


```

\newcommand{\definenuances}[3]{
\definecolor{#10}{#2}{#3}
\colorlet{#11}{#10!75} \colorlet{#12}{#10!50} \colorlet{#13}{#10!25}
\definenuances{orange1}{cmyk}{0.00,0.64,1.00,0.00}
% Puis plus loin dans le corps du document, les couleurs sont utilisables.
En une commande, quatre nuances sont définies : \textcolor{orange10}{$\blacksquare$},
\textcolor{orange11}{$\blacksquare$}, \textcolor{orange12}{$\blacksquare$} et
\textcolor{orange13}{$\blacksquare$}.
.....

```

En une commande, quatre nuances sont définies : , ,  et .

CODE 8.4 – Définition de commande à plusieurs arguments

Dans ce cas précis, `#10` signifie « l'argument n° 1 suivi du texte `0` », l'argument ne pouvant avoir un numéro supérieur à neuf.

8.2.2 L'espace

L'espace après une commande sans argument est absorbée par défaut par la commande, ce qui peut donner un texte erroné comme le montre l'exemple suivant où la commande `\LaTeX` se retrouve accolée au « peut » qui la suit.

Ce point peut se corriger en indiquant derrière la commande `{}` pour indiquer un argument vide. Pour simplifier cette gestion de l'espace, le paquet `xspace` met à disposition la commande `\xspace` traitant correctement cette espace après un nom de commande.

```

\newcommand{\MonLaTeX}{\LaTeX\xspace}
\LaTeX peut surprendre. Tout comme \LaTeX{} peut ne pas surprendre. De fait, s'il est
bien utilisé, \MonLaTeX ne surprend plus.
.....

```

\LaTeX peut surprendre. Tout comme \LaTeX peut ne pas surprendre. De fait, s'il est bien utilisé, \LaTeX ne surprend plus.

CODE 8.5 – Exemples d'utilisation de `\xspace`

8.2.3 Les mathématiques

Les nouvelles commandes traitant des mathématiques posent souvent une question de définition. Faut-il les appeler en les plaçant dans un environnement mathématique (`\$somme\$`) ou faut-il les appeler directement (`\somme`), leur définition incluant le passage en mode mathématique ? La commande `\ensuremath` simplifie ce point en traitant automatiquement l'éventuelle bascule en mode mathématique de son argument.

```
\newcommand{\somme}{\ensuremath{\sum_{k=0}^n x_k}}
Appeler \somme, $\somme$ ou bien encore \[\somme\] donne des résultats similaires
(à la présentation du dernier près).
```

Appeler $\sum_{k=0}^n x_k$, $\sum_{k=0}^n x_k$ ou bien encore

$$\sum_{k=0}^n x_k$$

donne des résultats similaires (à la présentation du dernier près).

CODE 8.6 – Définition de commande mathématique protégée

8.2.4 Les compteurs

LaTeX manipule, entre autres, des compteurs. Ceux-ci gèrent par exemple le numéro de page, la numérotation des chapitres. Il est possible de définir d'autres compteurs. Un compteur porte un *nom* (et n'est jamais précédé d'un `\`).

```
\newcounter{nom}
\setcounter{nom}{valeur entière}
\addtocounter{nom}{valeur entière}
```

CODE 8.7 – Gestion de compteurs

La commande `\setcounter` définit un nouveau compteur, `\setcounter` lui attribue la *valeur entière* tandis que `\addtocounter` lui ajoute la *valeur entière*. Pour afficher un compteur, il faut faire précéder le nom du compteur du préfixe `\the`. Ainsi le compteur gérant les numéros de page s'appelle `page`². Sa valeur est restituée par `\thepage`, en l'occurrence 61.

Voici un exemple d'utilisation : la création d'exemples numérotés automatiquement.

```
% Dans le préambule du document
\newcounter{nbex}
\setcounter{nbex}{0}
\newcommand{\exemple}{\addtocounter{nbex}{1}\textbf{Exemple \thenbex\ --- }}
% Dans le corps du document
\exemple Un test \par
\exemple Un autre test
```

Exemple 1 — Un test
Exemple 2 — Un autre test

CODE 8.8 – Exemple d'utilisation d'un compteur

8.2.5 Les dimensions

De façon parallèle aux compteurs, les quelques commandes suivantes permettent de gérer des longueurs avec deux différences importantes :

- le nom de chaque dimension commence bien ici par un `\` ;
- chaque *dimension* est exprimée avec son unité, parmi celles vues en page 8.

2. Étonnant, non ? Et que dire alors des compteurs `chapter`, `section`, `table`, `figure`...

```
\newlength{\nom}
\setlength{\nom}{dimension}
\addtolength{\nom}{dimension}
```

CODE 8.9 – Gestion de dimensions

Un exemple de recours à ces commandes est donné en 8.3.2 ainsi qu'en page 8.3.3.

8.2.6 Les boucles

Le paquet `multido` fournit une commande qui répète des actions en faisant si besoin varier un paramètre, autrement dit une boucle « for »³ pour \LaTeX . Elle a pour syntaxe :

```
\multido{indice=valeur initiale+incrément}{nombre}{actions}
```

CODE 8.10 – Boucle

Les *actions* sont répétées *nombre* fois. À chaque répétition, l'*indice* qui commence à la *valeur initiale* augmente de la valeur de l'*incrément* et il est utilisable. Voici deux exemples d'application de cette commande. Le premier fait écrire quatre fois `\LaTeX\dotfill` et une dernière fois `\LaTeX`. Le second affiche des boîtes de couleur⁴ dont la valeur teinte bleue varie en fonction de l'indice de la boucle, la valeur de l'indice étant affichée en blanc.

```
\multido{\Ia=1+1}{4}{\LaTeX\dotfill}\LaTeX \ \ \
\multido{\Ib=105+15}{11}{\colorbox[RGB]{0,0,\Ib}{\textcolor{white}{\textbf{\Ib}}}} }
.....
\LaTeX ..... \LaTeX ..... \LaTeX ..... \LaTeX ..... \LaTeX
105 120 135 150 165 180 195 210 225 240 255
```

CODE 8.11 – Exemples de boucle

Un exemple est également donné page 76.

8.2.7 Les conditions

Le paquet `ifthen` permet de générer des comportements conditionnels par la mise à disposition de la commande suivante :

```
\ifthenelse{condition}{commandes si condition vraie}{commandes sinon}
```

CODE 8.12 – Condition

La *condition* est exprimable par plusieurs commandes : par exemple, `\value{}{}` teste l'égalité de deux valeurs ; `\isodd{}` teste la parité d'un nombre. Ici est présenté un exemple qui utilise également le paquet `multido` :

3. Il existe à ce sujet un « while », non présenté ici, plus naturel en code \TeX .

4. La syntaxe indiquée ici pour `\colorbox` est une variante qui mélange déclaration de la couleur et utilisation de la couleur.

```
Ce chapitre, numérotée \thechapter, est \ifthenelse{\isodd{\thechapter}}{impair,
non?}{paire, n'est-ce-pas?} \\
\multido{\Ib=0+3}{15}{\ifthenelse{\equal{\Ib}{33}}{[Dites trente-trois] }{\Ib \ }}
```

```
Ce chapitre, numérotée 8, est paire, n'est-ce-pas ?
0 3 6 9 12 15 18 21 24 27 30 [Dites trente-trois] 36 39 42
```

CODE 8.13 – Exemple de test conditionnel

8.2.8 Pour aller plus loin

Quelques paquets permettent de pousser plus avant les développements possibles avec L^AT_EX :

- `calc` permet d'effectuer des calculs simples ;
- `ifpdf` permet de gérer des conditions basées sur le type de compilation retenue par l'utilisateur ;
- `xkeyval` permet de gérer des options dans les commandes sous forme d'une suite de terme « `option=valeur` » comme cela s'observe page 71. Il est à réserver aux développeurs motivés ;
- `comment` propose l'environnement `comment` qui encadre des zones où L^AT_EX considère tout le texte comme un unique commentaire.

8.3 La modification des réglages de L^AT_EX

8.3.1 Les libellés de division

Les noms des divisions comme « Chapitre » ou « Table des matières » sont en fait contenus dans des commandes (ici avec les valeurs une fois `babel` chargé avec l'option `frenchb`) :

➔Réfs
[13]485–549
[16]Tout
[31]255–278

Variable	Contenu par défaut
<code>\appendixname</code>	Annexe
<code>\bibname</code>	Bibliographie
<code>\chaptername</code>	Chapitre
<code>\contentsname</code>	Table des matières
<code>\indexname</code>	Index
<code>\listfigurename</code>	Table des figures
<code>\listtablename</code>	Liste des tableaux

TABLE 8.1 – Quelques noms de variables

En redéfinissant ces dernières, il devient possible de personnaliser sa présentation. La *redéfinition* se fait avec la commande `\renewcommand`. Une première manière de procéder revient à mettre cette commande juste après le `\begin{document}`. Une seconde manière permet de garder la redéfinition en préambule en utilisant une commande un peu plus avancée.

```
\addto\captionsfrench{\renewcommand{\contentsname}{Sommaire}} % Méthode 2
\begin{document}
\renewcommand{\contentsname}{Sommaire} % Méthode 1
```

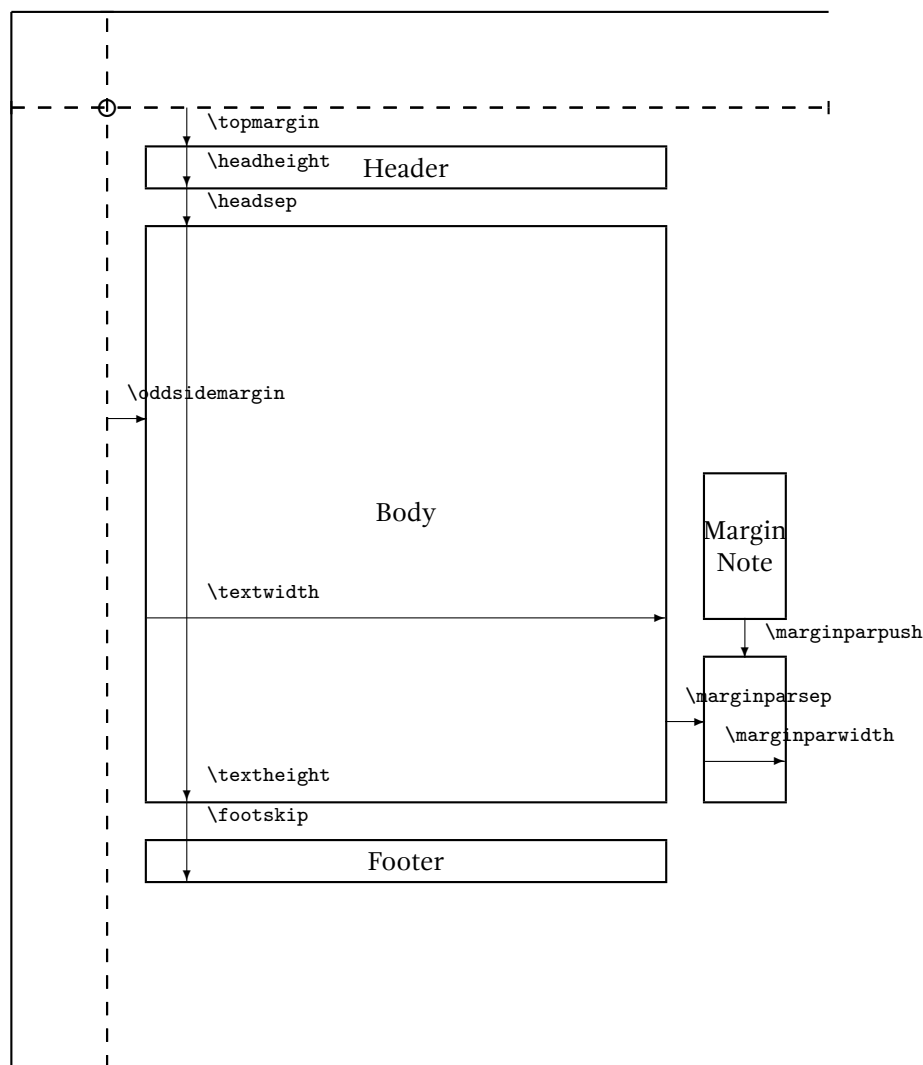
CODE 8.14 – Redéfinition du titre de la table des matières

8.3.2 Les dimensions de la page

Pour définir une page, \LaTeX utilise plusieurs variables représentant des dimensions. Parmi celles-ci se trouvent par exemple la hauteur de la page, la largeur des marges, la largeur du texte principal, la hauteur de l'en-tête.

Le paquet `layout` permet d'afficher schématiquement l'ensemble des dimensions d'une page en mettant à disposition la commande `\layout`. La page suivante utilise un paquet similaire, `layouts`, un peu plus étendu dans ses fonctionnalités.

The circle is at 1 inch from the top and left of the page. Dashed lines represent ($\backslash\text{hoffset} + 1\text{ inch}$) and ($\backslash\text{voffset} + 1\text{ inch}$) from the top and left of the page.



Actual page layout values.

$\backslash\text{paperheight} = 845.04684\text{pt}$	$\backslash\text{paperwidth} = 597.50787\text{pt}$
$\backslash\text{hoffset} = 0.0\text{pt}$	$\backslash\text{voffset} = 0.0\text{pt}$
$\backslash\text{evensidemargin} = 13.08827\text{pt}$	$\backslash\text{oddsidemargin} = 13.08827\text{pt}$
$\backslash\text{topmargin} = -26.08409\text{pt}$	$\backslash\text{headheight} = 14.0\text{pt}$
$\backslash\text{headsep} = 18.06749\text{pt}$	$\backslash\text{textheight} = 654.41338\text{pt}$
$\backslash\text{textwidth} = 426.79134\text{pt}$	$\backslash\text{footskip} = 25.29494\text{pt}$
$\backslash\text{marginparsep} = 7.0\text{pt}$	$\backslash\text{marginparpush} = 5.0\text{pt}$
$\backslash\text{columnsep} = 10.0\text{pt}$	$\backslash\text{columnseprule} = 0.0\text{pt}$
$1\text{em} = 10.0\text{pt}$	$1\text{ex} = 4.74\text{pt}$

FIGURE 8.1 – Une page et ses dimensions

En tant que variables, les dimensions d'une page peuvent être librement redéfinies par l'utilisateur à tout endroit dans le document. Pour ce document de cours, les décalages verticaux et horizontaux étaient à l'origine diminués de 30 points et la hauteur comme la largeur du texte

étaient augmentés de 60 points (pour garder l'aspect centré du texte)⁵ en utilisant la commande `\addtolength` :

```
\addtolength{\hoffset}{-30pt}
\addtolength{\textwidth}{60pt}
\addtolength{\voffset}{-30pt}
\addtolength{\textheight}{60pt}
```

CODE 8.15 – Modification de page par commande

Les modifications manuelles devraient cependant rester l'exception. Pour faire des modifications mieux encadrées, le paquet `geometry` constitue un excellent outil de travail : sur la base de quelques réglages, il étalonne les autres dimensions de la page. Ce qui suit illustre le cas de ce document. Il est demandé d'avoir une largeur de zone de texte de 15 cm et une hauteur de zone de texte de 23 cm. Le paquet `geometry` traite cette demande en modifiant toutes les variables de la page de façon à obtenir une présentation équilibrée.

```
\usepackage[body={15cm,23cm}]{geometry}
```

CODE 8.16 – Modification de page pour ce document

8.3.3 La personnalisation des en-têtes et pieds de page

Le paquet `fancyhdr` (pour « fancy header ») permet d'affiner très largement la présentation des en-têtes et pieds de page. Il propose en effet de découper les en-têtes et les pieds de page en trois zones : droite (R), centre (C) et gauche (L)⁶. Pour illustrer les principes de ce paquet sont présentés ici les choix effectués pour la présentation de ce document. Dans un premier temps, le paquet est chargé et il est demandé l'application du style paramétrable mis à disposition : `fancy`.

```
\usepackage{fancyhdr}
\pagestyle{fancy}
```

CODE 8.17 – Appels pour l'application du style `fancy`

L'en-tête fait apparaître ici des éléments liés au titre des chapitres ou des sections. Ceci demande quelques commandes spécifiques pour le stockage d'une part et la restitution d'autre part des éléments de titre.

En terme de stockage, lorsqu'il trouve une commande `\chapter{}` ou `\section{}`, L^AT_EX utilise respectivement les commandes `\chaptermark` et `\sectionmark` pour en retenir le contenu à destination des en-têtes et pieds de page. Ces deux commandes peuvent stocker de deux façons différentes l'information selon qu'elles utilisent la commande `\markboth{ }{ }` ou `\markright{ }`. `\markboth{ }{ }` stocke dans un premier groupe son premier argument et dans un second groupe son second argument. `\markright{ }` stocke uniquement dans le second groupe. Tant que, sur une page, aucun stockage n'est fait, les deux groupes contiennent par défaut les valeurs associées à la page précédente. Ils sont totalement remplacés par la première valeur stockée au premier appel sur une page.

En terme de restitution dans nos en-têtes et pieds de page, la commande `\leftmark` récupère pour une page donnée le *dernier* élément mis dans le premier groupe. La commande `\rightmark`, elle, récupère pour une page donnée le *premier* élément mis dans le second groupe.

5. On pourrait aussi définir directement la taille plutôt que de la modifier par rapport à l'existant.

6. Cette distinction peut aller plus en présentant séparément page impaire (O pour *odd*) et page paire (E pour *even*) si l'option générale `twoside` a été choisie pour le document.

Pour ce document, il faut stocker le nom du chapitre quand un chapitre est rencontré et le numéro de la section (obtenu avec la commande `thesection`⁷) suivi de son nom quand une section est croisée. Les commandes de stockage vont être les suivantes :

```
\renewcommand{\chaptermark}[1]{\markright{#1}}
\renewcommand{\sectionmark}[1]{\markright{\thesection\ #1}}
```

CODE 8.18 – Redéfinition des textes d’en-tête pour chapitre et section

La présentation de l’en-tête peut être maintenant travaillée. Tout d’abord, `\fancyhf{}` annule toutes les présentations existantes. La ligne suivante donne la définition du contenu de l’en-tête à droite (R) : en gras, le numéro de page obtenu avec `\thepage`. La dernière ligne définit le contenu de l’en-tête à gauche (L) : en l’occurrence, `\rightmark` qui va restituer, du fait des définitions de stockage et sa propre définition, l’information du premier chapitre ou la première section apparaissant sur la page et à défaut la même valeur que la page précédente. Pour information, si on avait souhaité intervenir les pieds de page, nous aurions utilisé la commande `\fancyfoot` avec le même principe.

```
\fancyhf{}
\fancyhead[R]{\bfseries\thepage}
\fancyhead[L]{\bfseries\rightmark}
```

CODE 8.19 – Contenu et présentation de l’en-tête

Les lignes suivantes modifient directement l’épaisseur des filets associés à l’en-tête et au pied de page, ce dernier étant effacé en lui donnant une épaisseur nulle. La dernière ligne corrige une légère anomalie de dimension que signale L^AT_EX sinon.

```
\renewcommand{\headrulewidth}{0.5pt}
\renewcommand{\footrulewidth}{0pt}
\addtolength{\headheight}{2pt}
```

CODE 8.20 – Gestion des filets et de l’espace de l’en-tête

La toute dernière ligne de cet exemple définit un style « `plain` » qui est vide, au cas où l’utilisateur souhaiterait pour une page particulière un en-tête vide et sans filet.

```
\fancypagestyle{plain}{\fancyhead{} \renewcommand{\headrulewidth}{0pt}}
```

CODE 8.21 – Définition d’un en-tête vide

8.3.4 La personnalisation de l’index

Un index sur plusieurs colonnes

Par défaut, l’index n’est pas un chapitre à part entière, il n’est pas mentionné dans la table des matières et il s’affiche sur une unique colonne qui le fait occuper rapidement beaucoup de pages. Voici un code (parmi d’autres) modifiant ces points en redéfinissant l’environnement `theindex`. Pour ce qui est de la gestion sur plusieurs colonnes, ici trois, il est fait usage du paquet `multicol` mettant à disposition l’environnement `multicols`.

7. De même, `\thechapter` affiche le numéro du chapitre et `\thepage` le numéro de page.


```
\renewenvironment{theindex}{%
  \chapter{\indexname}%
  \begin{multicols}{3}%
  \setlength\parindent{0pt} %
  \newcommand\item{\par\hangindent=40pt}}%
{\end{multicols}}
```

CODE 8.22 – Redéfinition de l'index

Comme vu en page 63, le nom de l'index est indiqué ici comme titre de chapitre⁸. L'affichage en trois colonnes est ensuite activé. Enfin, deux réglages sont effectués pour répondre aux règles de présentation de l'index. Ce dernier se base en effet en partie sur la commande `item` pour énumérer les différents principaux de l'index. Nous redéfinissons d'abord l'espace marquant l'indentation de début de paragraphe en le fixant à une valeur nulle (nous n'avons plus ici de paragraphes de texte). Puis nous redéfinissons la commande `item` comme une commande faisant commencer ou finir un paragraphe (`par`), le paragraphe qu'elle introduit ayant une indentation forcée de 40 pt dès sa deuxième ligne.

Un index avec un style personnalisé

Comme précisé en page 34, avec le programme `makeindex`, il est possible de recourir à des styles de présentation de l'index⁹, à l'image de ce qui est fait dans les bibliographies. Cette présentation se définit dans un fichier `.ist` (à placer dans le même répertoire que le fichier `.tex`). Ce fichier s'avère être un fichier texte contenant quelques mots clé pour lesquels il est indiqué le texte L^AT_EX qu'il faudra placer si l'élément du style associé au mot clé apparaît. Ce fichier doit respecter certaines règles d'écriture :

- `\n` correspond au retour à la ligne ;
- `\\` remplace systématiquement `\` lorsqu'une commande L^AT_EX doit être saisie.

Dans l'exemple qui suit, nous modifions les définitions par défaut de trois mots clés, ceux qui concernent les lettrines.

```
headings_flag 1
heading_prefix "{\bfseries "
heading_suffix "}\\nopagebreak \n"
```

CODE 8.23 – Lettrines simples pour un index

La première ligne indique que nous souhaitons des lettrines (0 sinon), la deuxième ligne que nous souhaitons l'insertion du code `{\bfseries` avant la lettrine (donc une mise en gras de la lettrine) et la dernière ligne que nous souhaitons l'insertion après la lettrine de la fin de la zone en gras (`}`) ainsi qu'une interdiction de saut de page à cet endroit avec `\nopagebreak`. Ainsi, la lettrine ne peut se retrouver seule en bas de page : elle sera renvoyée à la page suivante.

8.3.5 La gestion forcée des flottants

La gestion des flottants pour des documents assez courts peut paraître parfois quelque peu fantaisiste : une figure est le plus souvent supposée liée au texte qui l'entoure. Le paquet `floatrow`, parmi les éléments qu'il apporte dans la gestion des flottants, ajoute un nouveau type de disposition : `H`. Cette disposition impose l'affichage de la figure (ou d'un tableau) à l'endroit exact où elle apparaît dans le code. Il suffit donc d'utiliser l'instruction :

8. Dans le cas d'index multiples, cette ligne est à modifier car le nom de l'index n'est plus stocké dans cette variable. Le code nécessaire est placé dans le code source de ce document.

9. Voir <http://monbloginformatique.blogspot.fr/2010/03/latex-creer-un-joli-index.html> pour un autre exemple [7].

```
\begin{figure}[H]  
% figure  
\end{figure}
```

CODE 8.24 – Positionnement forcé d’une figure

8.3.6 Pour aller plus loin

Les différents ouvrages sur \LaTeX explicitent toujours de façon détaillée ces réglages, tout particulièrement les ouvrages de Bernard DESGRAUPES[13] et de Vincent LOZANO[24]. Parmi les paquets permettant des personnalisations, il convient de citer :

- `titlesec`, qui permet de modifier largement la présentation des différentes divisions du document : numérotation, espacements, format du texte ;
- `fncychap`, qui propose des présentations variées des chapitres ;
- `floatrow` et `caption` qui permettent de définir intégralement ses propres flottants, le second paquet gérant la partie mise en forme des légendes ;
- `enumitem`, qui propose des méthodes pour modifier les énumérations.

Enfin, le site de Vincent ZOONEKYND propose quelques codes pour des présentations alternatives des chapitres et sections : <<http://zoonek.free.fr/LaTeX/>>.

Chapitre 9

Les graphiques

Comme vu en page 53, le paquet `graphicx` permet d'insérer des images. Mais \LaTeX peut faire mieux : il est possible de le faire dessiner. Pour cela, deux grands paquets existent : `pstricks` et `tikz`. Le premier, plus ancien, dispose de plus de fonctionnalités diverses développées par différents amateurs au fil des années, tandis que le second est censé être plus précis pour ce qui est des définitions. Les deux sont ici présentés sur des cas similaires pour illustrer leur syntaxe respective.

9.1 PsTricks

Les éléments présentés ci-après se basent sur un paquet étendant largement les capacités graphiques rudimentaires de \LaTeX : `pstricks` de Timothy VAN ZANDT. Sont illustrées ici des bases de fonctionnement. Les lecteurs intéressés sont invités à se tourner vers la documentation accompagnant `pstricks`.

Ce paquet met à disposition un environnement pour traiter les figures ¹ :

```
\begin{pspicture}(largeur,hauteur)
% Figure pstricks
\end{pspicture}
```

CODE 9.1 – L'environnement de `pstricks`

Dans `pstricks`, l'unité de mesure est le centimètre. Cet environnement définit ici une zone de « `largeur cm` \times `hauteur cm` » dans laquelle se feront nos tracés.

9.1.1 Le placement

Avant de présenter les objets propres à `pstricks`, nous manipulons ici un objet bien connu : du texte. Comme tout objet graphique, il peut être positionné dans une figure avec la commande `\rput[position]{angle}(x,y){objet}`. La coordonnée (x,y) , présentée par une croix bleue dans l'exemple qui suit, sert de référence au placement de l'objet en suivant la consigne de *position* donnée (l'absence de position centrant l'objet sur cette coordonnée). L'exemple illustre le sens des positions `b`(bottom), `t`(top), `r`(right) et `l`(left). Enfin, l'*angle*, exprimé en degrés (sens trigonométrique), exprime l'angle de la rotation appliquée à l'objet.

1. Il est cependant possible de tracer *directement* dans le texte les graphiques... mais avec de nombreuses surprises à la clé en terme d'affichage.

➡Réfs
[13]—
[16]—
[31]325–348

```

\begin{pspicture}(10,2)
\psdots[dotstyle=BoldMul,linecolor=bleu6](1,0.5)(0.8,1.1)(8,1.1)(6,1)
\rput(1,0.5){1.\LaTeX}
\rput[b1](0.8,1.1){\resizebox{3.5cm}{!}{2.\textcolor{bleu5}{\LaTeX}}}
\rput{160}(8,1.1){\resizebox{!}{0.7cm}{3.\LaTeX}}
\rput[tr](6,1){\resizebox{!}{0.6cm}{\reflectbox{4.\LaTeX}}}
\end{pspicture}

```



CODE 9.2 – Placements et redimensionnements

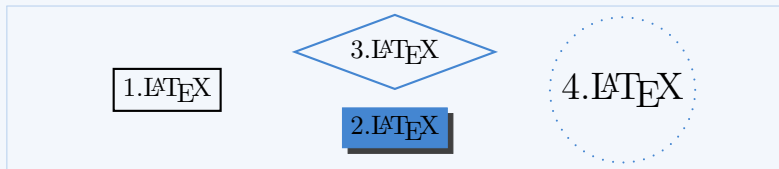
9.1.2 Les boîtes

Plusieurs commandes permettent d'entourer les objets. L'exemple qui suit illustre les boîtes `\psframebox`, `\psdiabox` et `\psciclebox`. Les versions étoilées des boîtes permettent de les remplir². L'exemple présente également la manière de saisir des options de présentation avec `pstricks` : en listant les options entre crochets sous la forme `option = valeur` et en séparant les différentes options par des virgules. Ces options permettent, entre autres, de gérer des paramètres comme l'épaisseur de trait, le type de trait, de remplissage ou la couleur.

```

\begin{pspicture}(10,2)
\rput(2,1){\psframebox{1.\LaTeX}}
\rput(5,0.5){\psframebox*[fillcolor=bleu6,shadow=true]{2.\LaTeX}}
\rput(5,1.5){\psdiabox[linecolor=bleu6]{3.\LaTeX}}
\rput(8,1){\psciclebox[linestyle=dotted,linecolor=bleu6]{\Large 4.\LaTeX}}
\end{pspicture}

```



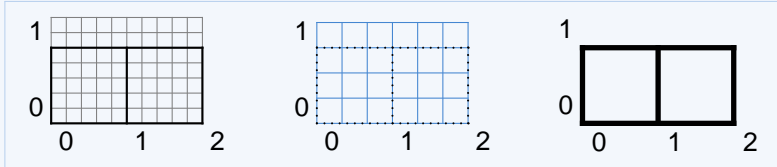
CODE 9.3 – Boîtes

9.1.3 Les grilles

Des grilles peuvent être créées aisément avec différents paramétrages. L'exemple qui suit illustre également une autre manière d'introduire la zone de tracé par l'environnement `pspicture` : la donnée des coordonnées du point inférieur gauche et du point supérieur droit du cadre.

2. Par défaut, le remplissage est blanc uni. Ainsi, l'étoile est ici strictement équivalente à l'option `fillstyle=solid`.

```
\begin{pspicture}(-0.5,-0.5)(9.5,1.5)
\psgrid(2,1.4)
\rput(3.5,0){\psgrid[griddots=10,subgriddiv=3,subgridcolor=bleu6](2,1.4)}
\rput(7,0){\psgrid[gridwidth=0.07,subgriddiv=0](2,1.4)}
\end{pspicture}
```

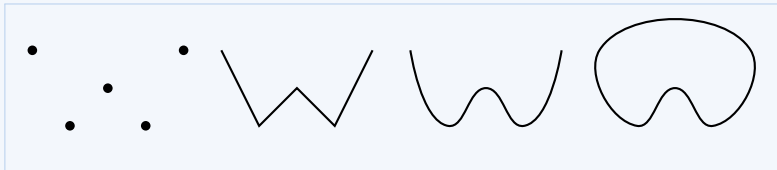


CODE 9.4 – Grilles

9.1.4 Les points et courbes

De la même manière, **psstricks** dispose de plusieurs commandes de base pour former des traits anguleux, curvilignes. L'exemple suivant les présente sans option.

```
\begin{pspicture}(-0.25,-0.5)(9.75,1.5)
\psdots(0,1)(0.5,0)(1,0.5)(1.5,0)(2,1)
\psline(2.5,1)(3,0)(3.5,0.5)(4,0)(4.5,1)
\pscurve(5,1)(5.5,0)(6,0.5)(6.5,0)(7,1)
\psccurve(7.5,1)(8,0)(8.5,0.5)(9,0)(9.5,1)
\end{pspicture}
```



CODE 9.5 – Points et courbes

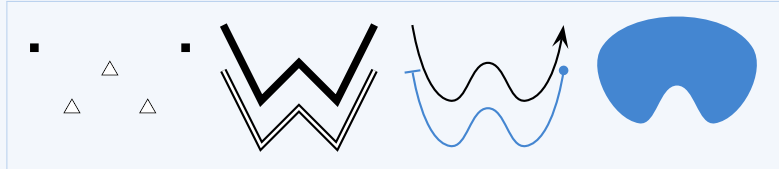
9.1.5 Les flèches et options

Points et courbes peuvent être modifiés en donnant leur épaisseur mais aussi leur forme. Dans le cas du trait, il s'agit de la forme aux extrémités, ce qui permet en particulier d'obtenir les flèches.

```

\begin{pspicture}(-0.25,-0.5)(9.75,1.5)
\psdots[dotstyle=square*](0,1)(2,1)
\psdots[dotstyle=triangle,dotsize=0.2](0.5,0.2)(1,0.7)(1.5,0.2)
\psline[doubleline=true](2.5,0.7)(3,-0.3)(3.5,0.2)(4,-0.3)(4.5,0.7)
\psline[linewidth=0.1](2.5,1.3)(3,0.3)(3.5,0.8)(4,0.3)(4.5,1.3)
\pscurve[linecolor=bleu6]{|-*}(5,0.7)(5.5,-0.3)(6,0.2)(6.5,-0.3)(7,0.7)
\pscurve[arrowscale=2]{->}(5,1.3)(5.5,0.3)(6,0.8)(6.5,0.3)(7,1.3)
\psccurve*[linecolor=bleu6](7.5,1)(8,0)(8.5,0.5)(9,0)(9.5,1)
\end{pspicture}

```



CODE 9.6 – Options sur points et courbes

9.1.6 Vers les mathématiques

La combinaison des éléments cités auparavant va permettre d'obtenir des graphiques mathématiques classiques : les flèches formeront les axes, les courbes la plupart des graphiques à tracer, le texte ajoutant des éléments comme les légendes des axes ou le nom des courbes. En général, il convient d'organiser les données numériques à tracer avec des logiciels permettant de concaténer des données en un format accessible à **pstricks**. En voici un exemple, obtenu avec des coordonnées mises sous forme de $(x_1, y_1)(x_2, y_2) \dots (x_n, y_n)$. Le code détaillé de cette figure est placé dans le fichier **3oat.tex** accompagnant le fichier du cours.

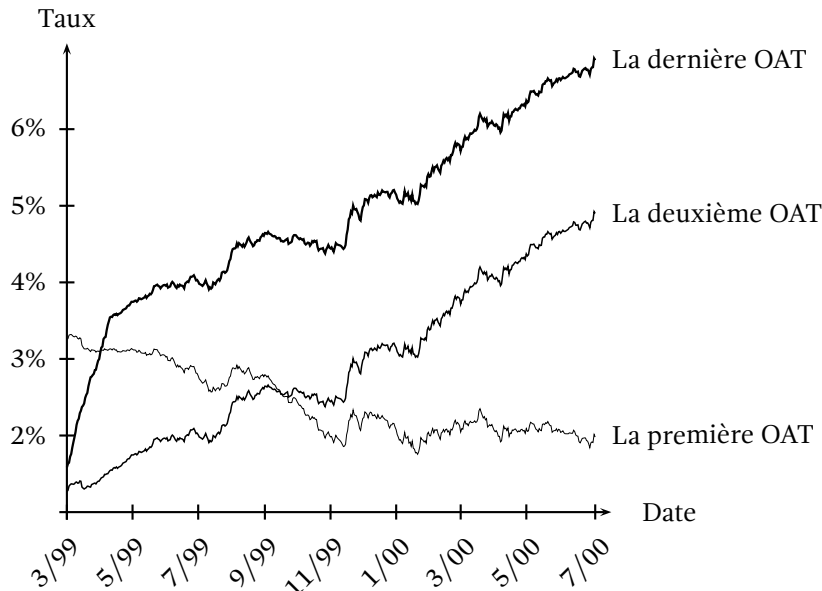


FIGURE 9.1 – Quelques OAT

9.1.7 Le tracé de fonction mathématique

Le paquet **pst-plot**, un des paquets spécialisés se basant sur **pstricks**, contient quelques commandes de tracé non négligeables. La plus simple, **\psaxes**, trace des axes à nos figures.

La notation des fonctions

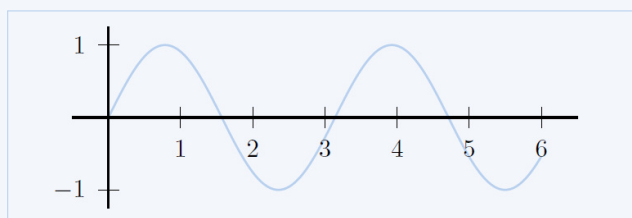
L'appel du paquet `pst-plot` permet d'utiliser en particulier les notations mathématiques usuelles avec l'option `algebraic=true` à placer parmi les options des commandes `\psplot` et `\parametricplot`, comme l'illustrent les exemples qui suivent. Sans cette option, les notations doivent se faire selon une notation polonaise inversée³.

La notation⁴ travaille en radian et non en degré. Dans le cas d'une commande paramétrique, la variable muette est `t` et il faut indiquer une séparation avec un `|` entre les deux fonctions paramétriques du `\parametricplot`.

Une fonction classique

Voici donc l'exemple de la fonction $\sin(2x)$.

```
\begin{pspicture}(-1,-1.25)(7,1.25)
\psplot[plotstyle=curve,plotpoints=199,linewidth=bleu8,algebraic=true]{0}{6}{sin(2*x)}
\psaxes(0,0)(-0.5,-1.25)(6.25,1.25)
\end{pspicture}
```



CODE 9.7 – Une sinusoïde

L'option `plotpoints` indique de plus le nombre de points à calculer pour tracer la courbe. Par ailleurs, l'option `plotstyle` permet de préciser avec la valeur `curve` que le tracé de la fonction se fait par courbe, autrement dit qu'il est lissé (il pourrait se faire par ligne avec la valeur `line` ou par de simples points avec `dots`).

Une fonction paramétrique

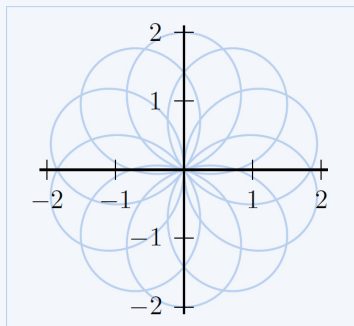
Pour des fonctions paramétrées, il faut mettre à la suite les fonctions de chaque coordonnées l'une après l'autre, \LaTeX repérant les deux fonctions du fait de la notation polonaise inversée. Voici l'exemple d'une rosace avec `\parametricplot` :

$$\begin{cases} x(t) &= 2 \sin 4t \cos 5t \\ y(t) &= 2 \cos 4t \cos 5t \end{cases}$$

3. Cette notation n'utilise pas la notion de parenthèses. Pour cela, les opérateurs se placent après les opérandes sur lesquels ils s'appliquent. Plus exactement, les termes se lisent successivement : si un terme est un opérande, il est stocké. Si c'est un opérateur, il s'applique aux derniers opérandes stockés.

4. Voir par exemple http://ww2.ac-poitiers.fr/math/spip.php?article134&debut_page=1.

```
\begin{pspicture}(-2.15,-2.15)(2.15,2.15)
\parametricplot[plotstyle=curve,plotpoints=199,linecolor=bleu8,algebraic=true]{0}{6.3}
{2*(sin(4*t))*(cos(5*t))|2*(cos(4*t))*(cos(5*t))}
\psaxes(0,0)(-2.1,-2.1)(2.1,2.1)
\end{pspicture}
```



CODE 9.8 – Une rosace

9.1.8 Exemples capillotractés

Un histogramme

Les commandes de `pstricks` peuvent être combinées encore plus fortement, jusqu'à définir un ensemble de commandes dédiées à des traitements particuliers. Il faut ici combiner utilisation de `pstricks` et programmation.

Voici, pour illustrer ce point, un histogramme obtenu avec un paquet inconnu car fait maison : `histogra`⁵.

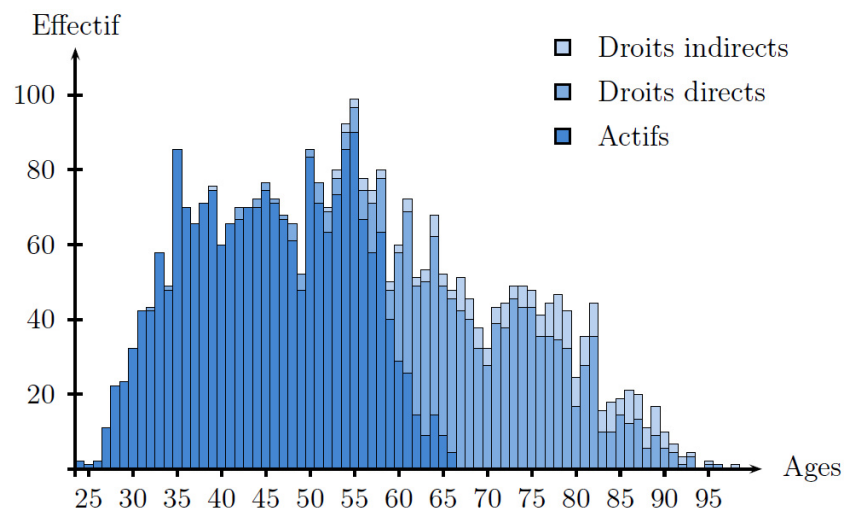


FIGURE 9.2 – Pyramide des effectifs

5. Ce paquet est très peu documenté et encore moins développé pour servir partout... il n'a servi que pour réaliser quelques figures dans un document bien précis.

Une famille de courbes

Ce qui suit utilise un mélange de commandes tirées des paquets `pstricks` et `multido`. Est tracé ici une famille de figures paramétrées définie par le système suivant :

$$\begin{cases} x(t) &= 0,9 \cos \alpha t \\ y(t) &= 0,9 \sin \beta t \end{cases} \quad \text{avec } (\alpha, \beta) \in [1..5] \times [1..6]$$

Pour faire ce tracé, une commande un peu plus générale est définie. Dans celle-ci, `multido` permet de faire deux boucles imbriquées (une sur α , l'autre sur β) et `pst-plot` permet de tracer la figure voulue.

```
\newcommand{\figu}[1]{
\begin{figure}[H]
\begin{center}
\fbbox{
\begin{pspicture}(12.3,15.2)
\multido{\Na=1.1+2.5,\Ia=1+1}{5}{
\multido{\Nb=13.9+-2.5,\Ib=1+1}{6}{
\rput{0}(\Na,\Nb){
\parametricplot[plotstyle=curve,plotpoints=199,linecolor=bleu6,
algebraic=true]{0}{6.2832}{\Ia}{\Ib}
\rput{0}(0,-1.2){\textsf{\footnotesize (\Ia,\Ib)}}}
}}
\end{pspicture}}
\caption{Exemples de fonctions paramétriques}
\end{center}
\end{figure}}
```

CODE 9.9 – Construction d’une famille de courbes

Ainsi, l’utilisation de la commande s’est limitée à l’appel qui suit. Cependant, en terme de présentation, cette commande devrait disposer de plus de paramètres que les formules. Sans cela, la légende, le nombre de figures tracées et leur précision ne peuvent être modifiés aisément.

```
\figu{0.9*cos(t*\Ia)|0.9*sin(t*\Ib)}
```

CODE 9.10 – Appel de la commande pour les courbes

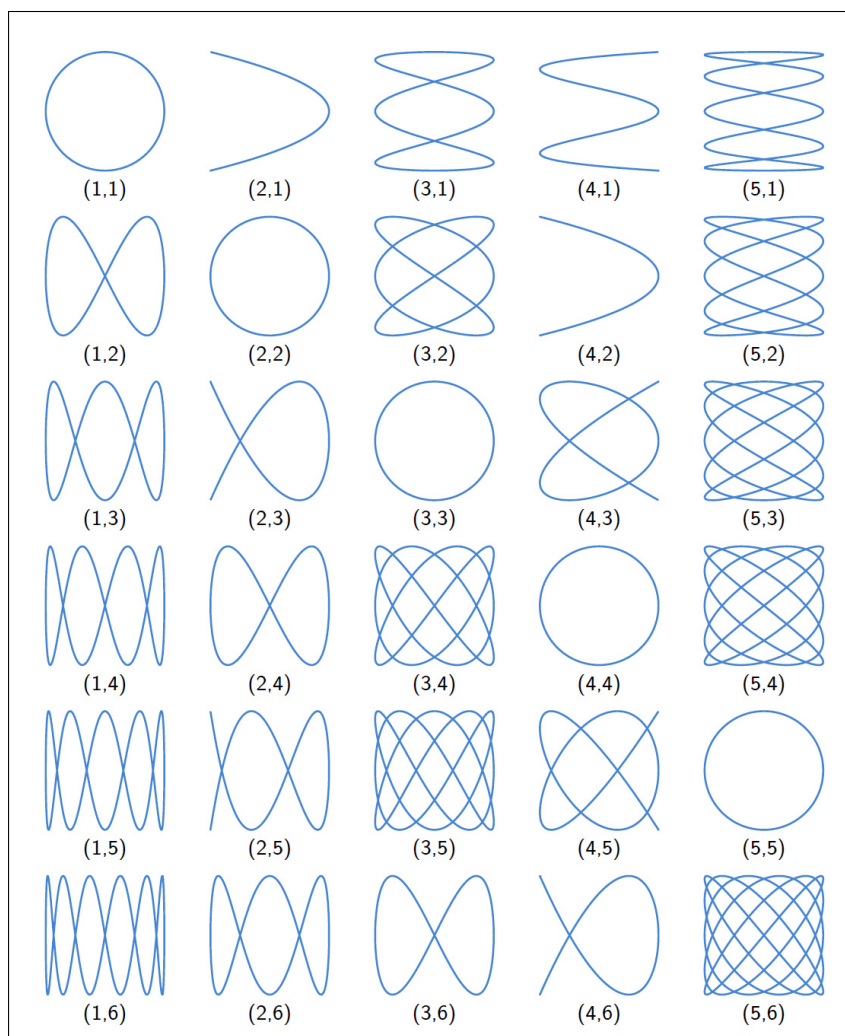


FIGURE 9.3 – Exemples de fonctions paramétriques

9.2 Tikz

Le paquet `tikz` est plus récent que le paquet `pstricks`. Par rapport à ce dernier, il présente deux avantages :

- il dispose généralement de plus grandes possibilités de paramétrages ;
- il fonctionne avec les différentes chaînes générant directement des fichiers `pdf`, ce qui en fait la solution graphique pour les fichiers image `jpg` mais aussi avec les variantes de \LaTeX que sont \XeTeX et \LuaTeX .

En contrepartie, il demande un peu plus de temps pour être utilisé. Par ailleurs, sur des graphiques nécessitant des calculs, il est plus lent que `pstricks`, ce dernier bénéficiant des mécaniques propres au langage postscript.

Ce paquet met à disposition un environnement pour traiter les figures :

```
\begin{tikzpicture}[options]
% Figure tikz
\end{tikzpicture}
```

CODE 9.11 – L’environnement de `tikz`

Cet environnement ne nécessite pas, contrairement à celui de `pstricks` de dimension : `tikz` va déterminer les dimensions de la zone à afficher pour qu’elle contienne et affiche l’ensemble des objets tracés⁶. Qui plus est, `tikz` prend en compte d’éventuelles *options* : ces dernières vont s’appliquer à l’ensemble des objets décrits par la suite, sauf règle explicite contraire présente dans un objet ou une classe d’objets.

9.2.1 Le placement par les nœuds

Sous `tikz`, les éléments sont placés sur un nœud (*node*). Un nœud peut être défini par des coordonnées ou de façon relative à d’autres points. Il peut également être appelé par son nom.

```
\begin{tikzpicture}[outer sep=0cm,inner sep=0cm]
\node (aa) at (1,0.5) {1.\LaTeX};
\node (ab) at (0.8,1.1) {};
\node[anchor=south west] at (ab) {\resizebox{3.5cm}{!}{2.\textcolor{bleu5}{\LaTeX}}};
\node[rotate=160] (ac) at (8,1.1) {\resizebox{!}{0.7cm}{3.\LaTeX}};
\node (ad) at (6,1) {};
\node[anchor=north east,outer sep=0cm,inner sep=0cm] at (ad)
{\resizebox{!}{0.6cm}{\reflectbox{4.\LaTeX}}};
\end{tikzpicture}
```



CODE 9.12 – Placements et redimensionnements

9.2.2 Les boîtes ou formes

Différentes formes sont prédéfinies sous `tikz`⁷. Se retrouve ici la logique de préciser le tracé du pourtour, le remplissage de la forme, la présence d’une ombre, que ce soit pour la couleur, l’épaisseur, les motifs...).

De nombreuses valeurs sont disponibles sous forme de mots clés. Par exemple, `thick` utilisé dans l’exemple définit un trait de 0,8pt d’épaisseur.

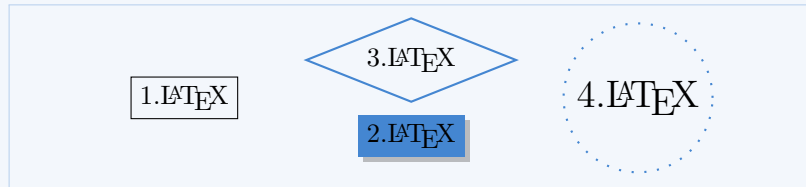
6. Ce comportement peut se piloter. Il suffit de placer des nœuds délimitant une zone plus large pour forcer `tikz` à suivre d’éventuels besoins de présentation.

7. en particulier avec la bibliothèque *shape* qui définit par exemple le losange utilisé dans l’exemple.

```

\begin{tikzpicture}
\node[rectangle,draw] at (2,1) {1.\LaTeX};
\node[rectangle,fill=bleu6,drop shadow] at (5,0.5) {2.\LaTeX};
\node[diamond,aspect=2.5,draw=bleu6,thick] at (5,1.5) {3.\LaTeX};
\node[circle,draw=bleu6,loosely dotted,thick] at (8,1) {\Large 4.\LaTeX};
\end{tikzpicture}

```



CODE 9.13 – Boîtes

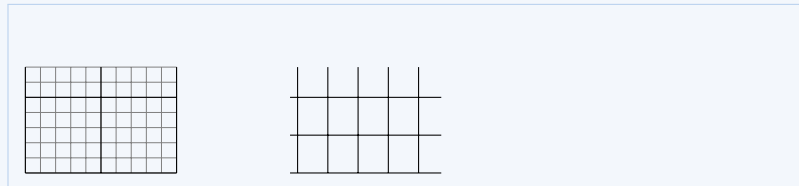
9.2.3 Les grilles

Les grilles se présentent sous une forme un peu différente de celle vue dans `pstricks`. En effet, la fonctionnalité la plus courante ne propose pas la numérotation des axes; de même la grille se base sur les coordonnées du plan ce qui fait que la seconde grille ci-dessous ne commence pas par un filet vertical.

```

\begin{tikzpicture}%[outer sep=0cm,inner sep=0cm]
\node (a) at (0,0) {}; \node (b) at (10,2) {};
\draw[step=.2cm,gray,very thin] (0,0) grid (2,1.4);
\draw[step=1cm] (0,0) grid (2,1.4);
\draw (3.5,0) grid[xstep=0.4cm,ystep=0.5cm] (5.5,1.4);
\end{tikzpicture}

```



CODE 9.14 – Grilles

Par contre, lorsqu'il s'agit de traiter la représentation de données, `tikz` dispose de commandes avancées pour traiter ce point.

9.2.4 Les points et courbes

De la même manière, `pstricks` dispose de plusieurs commandes de base pour former des traits anguleux, curvilignes. L'exemple suivant les présente sans option.

```

\begin{tikzpicture}%[outer sep=0cm,inner sep=0cm]
\node (a) at (0,0) {}; \node (b) at (10,2) {};
\fill (0,1) circle (2pt);
\fill (0.5,0) circle (2pt);
\fill (1,0.5) circle (2pt);
\fill (1.5,0) circle (2pt);
\fill (2,1) circle (2pt);
\draw (2.5,1) -- (3,0) -- (3.5,0.5) -- (4,0) -- (4.5,1);
\draw plot[smooth] coordinates{(5,1) (5.5,0) (6,0.5) (6.5,0) (7,1)};
\draw plot[smooth cycle,tension=0.8] coordinates{(7.5,1) (8,0) (8.5,0.5) (9,0) (9.5,1)};
\end{tikzpicture}

```



CODE 9.15 – Points et courbes

✂ Présentation à poursuivre ✂

Chapitre 10

Les polices de caractères avec X_YLaTeX

La méthode de gestion d'autres polices de caractères présentée page 15 est limitée dans ces possibilités. Il en existe une autre, plus complexe mais *très* efficace et qui passe par l'utilisation de variantes récentes de TeX : X_YTeX (et X_YLaTeX) qui rend plus intuitive l'utilisation de polices de caractères en donnant accès direct à l'ensemble des fontes utilisées par Windows ou MacOS.

10.1 Modifications associées à X_YLaTeX

Pour utiliser X_YLaTeX, il faut vérifier que le préambule est bien cohérent avec la version présentée en page 12. La déclaration d'un jeu particulier de polices de caractères se fait ensuite par la commande suivante.

```
\newfontfamily\mapolice[
  BoldFont = policegrasse.ttf,
  ItalicFont = policeitalique.ttf,
  BoldItalicFont = policeitaliquegrasse.ttf
]{policenormale.ttf}
```

CODE 10.1 – Création d'une famille de polices de caractères

La commande est donnée dans une version détaillée permettant de bénéficier de certaines fonctionnalités courantes. X_YLaTeX peut attendre d'une police de caractères qu'elle présente quatre versions : une normale, une variante grasse, une variante italique et une variante italique grasse. Certaines polices de caractères présentent naturellement ces quatre variantes, d'autres pas. Aussi la commande peut donc être parfois moins détaillée.

Les noms indiqués en italique correspondent aux noms des fichiers de polices de caractères nous intéressant et présents dans le répertoire usuel du système d'exploitation¹ : ce peut être des fichiers .ttf ou .otf.

Pour utiliser ces polices dans le document, il suffit d'utiliser la nouvelle bascule mise à disposition par la commande, ici `\mapolice`.

Dans le cas où il faut modifier les polices de caractères utilisées par défaut par LaTeX, il est fait usage d'une autre commande à la syntaxe similaire.

1. Il s'agit de `C:\Windows\Fonts` pour Windows.

```
\setmainfont[Path=fontes/,  
BoldFont = policegrasse.ttf,  
ItalicFont = policeitalique.ttf,  
BoldItalicFont = policeitaliquegrasse.ttf  
{police normale.ttf}
```

CODE 10.2 – Mise en valeur par défaut d’une famille de polices de caractères

La commande est ici présentée avec l’option `Path` qui permet d’indiquer un chemin où trouver les fichiers des polices de caractères. Cette commande gère la fonte principale et ses variantes. Si une variante sans serif existe, elle peut être également déclarée avec `\setsansfont` avec les mêmes options.

10.2 Pour aller plus loin

Il est ici recommandé de consulter la documentation de `fontspec` pour mieux appréhender les possibilités de ce paquet. Un point doit être cependant conservé à l’esprit : la manipulation de polices de caractère devrait être limitée pour des documents longs car ceci peut involontairement conduire à des documents relativement peu lisibles et souvent contraires à certains usages typographiques.

Plus largement, le logiciel FontForge², libre, permet d’éditer des polices de caractères, ne serait-ce par exemple que pour créer de façon quasi-automatique des petites capitales sur des polices de caractères existantes.

2. <<http://fontforge.github.io/en-US/>>.

Chapitre 11

Un fichier d'extension

Un paquet ou extension est un fichier d'extension `.sty` regroupant des ensembles de définitions permettant de modifier et d'élargir les traitements qu'effectue \LaTeX . Ce chapitre décrit la structure d'un tel fichier.

11.1 Déclarations

Afin d'être traité correctement par \LaTeX et par les différentes mécaniques associées à la distribution, ce fichier contient quelques commandes de description du paquet, placées en début de fichier :

```
\NeedsTeXFormat{LaTeX2e}  
\ProvidesPackage{nompaket}[aaaa/mm/jj version et autres infos]
```

CODE 11.1 – Déclaration d'un paquet

Le format de la date (*aaaa/mm/jj*) est normalisé mais le reste de la description est libre.

11.2 Options

11.2.1 Déclaration des options

L'appel d'un paquet peut se faire avec des options, comme vu en ??, page ??. Chacune des options est en fait un code présent dans le fichier du paquet. Ce *code* est rattaché au terme *nomoption*, le nom de l'option, et est à déclarer par la commande suivante :

```
\DeclareOption{nomoption}{code}
```

CODE 11.2 – Déclaration et définition d'une option

11.2.2 Traitement des options

Une des options ainsi déclarée peut être considérée comme l'option par défaut. Dans ce cas, après avoir déclaré les différentes options, la commande suivante permet d'exécuter systématiquement le code des options listées (séparées par des virgules).


```
\ExecuteOptions{liste d'options}
```

CODE 11.3 – Exécution d'options listées

Pour traiter le choix d'options fait par l'utilisateur et exécuter les codes associés, il faut alors placer la commande suivante.

```
\ProcessOptions \relax
```

CODE 11.4 – Exécution des options appelées par l'utilisateur

11.2.3 Transmission d'options

Dans le code contenu dans une commande `\DeclareOption`, il est possible d'indiquer des options à transmettre à un paquet appelé par la suite dans l'extension.

```
\PassOptionsToPackage{liste d'options}{paquet}
```

CODE 11.5 – Transmission d'options

11.3 Appel d'autres paquets

Certaines définitions ou fonctionnalités demandent à charger d'autres paquets. Pour cela, il ne faut pas utiliser la commande `\usepackage` mais une commande spécifique aux fichiers d'extension :

```
\RequirePackage[options]{paquet}
```

CODE 11.6 – Appel de paquet

11.4 Codes particuliers

Certains éléments d'un paquet peuvent demander des exécutions à des moments particuliers. Quelques commandes sont associées à trois moments de la compilation : à la fin de l'exécution du code du paquet, au début du corps du document ou à fin du document. Elles stockent le code pour exécution au moment qu'elles indiquent.

```
\AtEndOfPackage{code}  
\AtBeginDocument{code}  
\AtEndDocument{code}
```

CODE 11.7 – Stockage de code pour exécution à des moments spécifiques

Ces commandes peuvent être utilisées à plusieurs reprises : les codes qu'elles contiennent s'empilent alors en vue de l'exécution au moment associé.

Par ailleurs, la commande `\AtBeginDocument` est considérée comme appartenant au préambule du document et non au corps. Elle ne peut donc contenir de texte.

11.5 Exemple

Voici un exemple de code d'un paquet `demo`.

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{demo}[1902/01/17 V1.0 Quelques chargements]

% Déclaration des options
\DeclareOption{default}{%
\newcommand{\signature}{\textit{Signature}}}%
}

\DeclareOption{perso}{%
\newcommand{\signatureab}{Ambrose \textsc{Bierce}}}%
\PassOptionsToPackage{dvipsnames}{xcolor}%
}

% Traitement des options
\ExecutesOption{default}
\ProcessOptions \relax

% Appels de paquets et définitions communes
\RequirePackage{xcolor}
\RequirePackage{xspace}
\newcommand{\MonLaTeX}{\LaTeX\hspace}
```

CODE 11.8 – Exemple

Ce paquet dispose de deux options : `default` et `perso`. La première option est exécutée par défaut avec la commande `\ExecutesOption` et crée une commande `\signature`. La seconde option n'est exécutée que si le paquet `demo` est appelé avec l'option `perso`. Est alors définie une commande `\signatureab`. De plus, il est demandé de passer une option complémentaire au paquet `xcolor` lorsque ce dernier sera appelé : `dvipsnames`.

Une fois les options traitées, ce paquet appelle le paquet `xcolor` et le paquet `xspace`. L'option `perso` aura ici modifié le premier appel en lui ajoutant l'option `dvipsnames`.

Enfin, une fonction `\MonLaTeX` est définie.

11.6 Pour aller plus loin

Il est également possible de créer des classes de document (fichiers `.cls` sur des principes très similaires. Point non signalé ci-dessus, il est également possible d'inclure des textes d'erreur à afficher si une erreur survient. Ces deux éléments sont décrits dans cette documentation en anglais : <http://www.latex-project.org/guides/clsguide.pdf>.

Il est également possible de faire un travail plus complet en créant des fichiers qui génèrent à la fois le code du paquet (ou de la classe), la documentation générale et la documentation du code. La méthode est présentée dans un document en français : <http://yvon-henel.fr/texnicien/docs/dtxtut-fr/dtxtut-fr.pdf>.

Chapitre 12

PythonTeX

12.1 Brève présentation du langage

✖ À compléter ✔

12.2 Calcul scientifique

✖ À compléter ✔

12.3 Exemples d'utilisation

✖ À compléter ✔

12.4 Pour aller plus loin

✖ À compléter ✔

Quatrième partie

Annexes

Annexe A

Installation

Les installations présentées ici sont sous Windows. Pour une installation Mac, il convient de consulter le document de Fabien CONUS et de Franck PASTOR[12].

L'installation de \LaTeX est très fortement marquée par l'aspect modulaire de ce programme : beaucoup de fichiers sont créés et dépendent les uns des autres. Par ailleurs, \LaTeX crée par défaut des documents affichables mais non imprimables. L'installation de certains programmes complémentaires permet de convertir ces documents dans des formats imprimables.

A.1 Une installation compacte : $\text{USB}\text{\TeX}$

L'installation retenue pour le cours est $\text{USB}\text{\TeX}$ du fait de sa simplicité d'utilisation. Il s'agit d'un regroupement de plusieurs programmes de taille réduite : la distribution (voir ci-dessous) $\text{MiK}\text{\TeX}$, les programmes Ghostscript, GSview et l'éditeur Texmaker.

A.1.1 Installation

Le fichier exécutable d'un peu plus de 150 Mo est téléchargeable par le biais du site de Framasoft¹. L'exécution de ce fichier le fait se décompresser dans le répertoire demandé par la fenêtre qui s'affiche alors. Ceci fait, il faut se placer dans le répertoire nouvellement créé par l'installation² et cliquer sur le raccourci nommé [Texmaker](#) pour pouvoir utiliser l'éditeur Texmaker.

A.1.2 Maintenance

Cas usuel

L'installation initiale faite ci-dessus inclut un nombre restreint de paquet. Si, lors d'une compilation, un paquet semble manquant pour l'installation, Texmaker vous propose de l'installer directement par téléchargement en ligne (en passant par le logiciel de maintenance de $\text{MiK}\text{\TeX}$). Ceci évite de se poser des questions concernant les paquets, leur installation et leur maintenance.

Cas moins usuel

L'installation de paquet peut aussi se faire de façon plus manuelle en allant directement installer les paquets souhaités. Pour le faire, il faut lancer le raccourci [MiKTeX](#) présent dans le répertoire créé par l'installation.

Ceci fait apparaît une nouvelle icône dans la barre des tâches. En cliquant avec le bouton droit dessus et en sélectionnant [MiKTeX Package Manager](#) se lance alors le gestionnaire de paquet. Il peut mettre un certain temps avant de réagir car il compare les paquets présents et les paquets

1. <<http://www.framasoft.net/article4641.html>>. S'y trouve également le mode d'emploi résumé ci-dessus.

2. Ce répertoire est nommé [USBTeX](#)- suivi du numéro de version d' $\text{USB}\text{\TeX}$.

disponibles. Une fois la liste des paquets affichée, il suffit de sélectionner ici n'importe quel paquet et de cliquer sur l'icône + pour l'installer.

Cas improbable mais vrai

Il peut arriver que des problèmes de configuration de connexion empêche d'utiliser ce système directement. Si la connexion internet est fonctionnelle, il est possible de contourner cette difficulté en récupérant soi-même les fichiers que va chercher le gestionnaire de paquets sur des dépôts (*repository* en anglais) et en les stockant dans un répertoire dédié sur son ordinateur.

Une fois connue l'adresse d'un dépôt³, il faut alors y récupérer et placer dans un unique répertoire de son ordinateur les éléments suivants :

- les fichiers dont le nom commence par [miktex-zzdb](#) ;
- les fichiers des paquets qui vous intéressent.

Le premier ensemble de fichiers sert à indiquer la liste des paquets du dépôt au gestionnaire de paquet. En l'absence de ces fichiers, le gestionnaire de dépôt refusera de prendre en compte votre répertoire.

Dans le gestionnaire de paquet MiKTeX, il faut alors aller dans le menu [Repository](#) puis [Change Package Repository](#), sélectionner alors la ligne ... [from a directory](#) puis [Suivant](#), sélectionner le répertoire où sont vos paquets et cliquer enfin sur [Terminer](#).

A.1.3 La vérification orthographique

L'installation sur clé présente une coquille pour ce qui est du correcteur orthographique. Il faut modifier le chemin présent dans le menu [Options](#), [Configurer Texmaker](#) puis [Editeur](#). Le chemin de la ligne [Dictionnaire](#) est à modifier par le biais de la petite icône à côté de la zone de saisie du chemin, en cherchant le répertoire où est installé USBTeX ([USBTeX-1.7](#)) puis en allant dans le sous-répertoire [programs](#) puis [Texmaker](#) et sélectionner alors le fichier [fr_FR.dic](#).

A.2 Une installation complète

Le côté pratique d'USBTeX cache les liens entre les différents programmes. Est donc décrite ici une installation complète⁴ sous Windows. Celle-ci demande de récupérer trois ensembles :

- une **distribution** de \LaTeX , autrement dit \LaTeX et de nombreux programmes et fichiers satellites (paquets, documentations, formats...);
- Ghostscript et GSview qui gèrent le postscript et les fichiers [.ps](#) et [.eps](#) ;
- un éditeur \LaTeX , point vu en section [A.3](#).

A.2.1 \LaTeX

La distribution retenue ici de \LaTeX est la \TeX Live 2014, soumise à la licence GNU⁵. La dernière distribution de la \TeX Live est disponible sur <http://www.tug.org/texlive/>. Plusieurs modes d'installation sont possibles :

- en téléchargeant un fichier d'installation qui récupère les différents fichiers nécessaires sur le web. Il se trouve sur le site en suivant le lien [Download](#) puis le lien [install-tl.zip](#). Une fois ce fichier décompressé, il faut exécuter le programme [install-tl.bat](#), qui affiche la fenêtre de la figure [A.1](#) ;

3. Par exemple <ftp://ftp.dante.de/pub/tex/systems/win32/miktex/tm/packages/>. Une liste de dépôts est disponible à l'adresse <http://miktex.org/pkg/repositories>.

4. Parmi d'autres possibles telle http://www.xmlmath.net/doculatem/install_miktex.html.

5. Licence qui gère le logiciel libre.

- en téléchargeant le DVD d'installation présent sur le site⁶. Ce fichier doit alors être gravé ou monté sur un lecteur virtuel. Une fois lancé, il faut lancer le fichier `install-tl-windows.bat` qui affiche le même menu qu'à la méthode précédente.

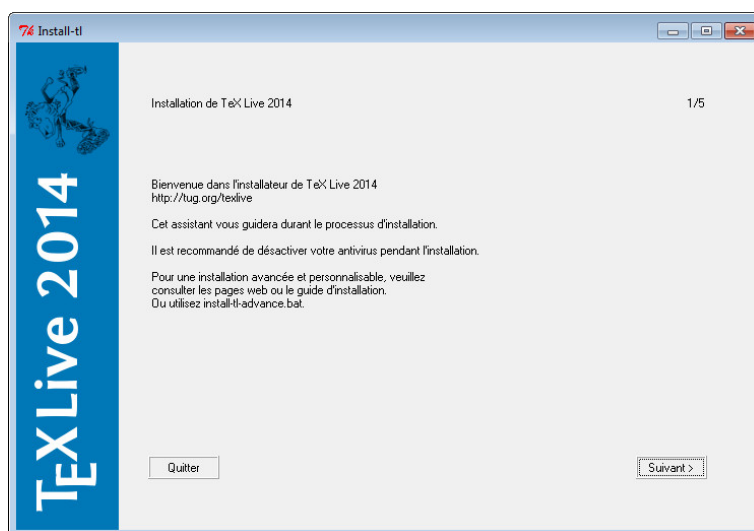


FIGURE A.1 – Page d'accueil de TeXLive 2014

Les différents choix usuels de configuration sont récapitulés dans le tableau suivant à raison d'une ligne par fenêtre d'installation.

Fenêtre	Action
1/5	Suivant
2/5	Répertoire C:\TeXLive\2014. Suivant
3/5	Suivant
4/5	Installer
5/5	Attendre et quitter

TABLE A.1 – Étapes d'installation de la TeXlive 2014

Cette partie est assez longue, surtout si c'est la méthode du fichier d'installation qui est retenue : les différents fichiers constitutifs de l'installation seront tous téléchargés. Dans la mesure où il y a beaucoup de fichiers de taille variable, la durée de téléchargement est elle-même délicate à mesurer. Par expérience, il faut compter deux bonnes heures.

Avec la méthode par DVD, le temps d'installation passe environ à une demi-heure.

A.2.2 Ghostscript et GSview

Ghostscript

Ghostscript permet de travailler avec des fichiers postscript ou `ps` ainsi que les images au format `eps`. Avec le programme `ps2pdf` compris dans les programmes fournis avec Ghostscript, il devient possible de convertir un fichier `ps` en `pdf`.

Une fois le programme dans sa version libre non commerciale (nommée GPL ou *General Public License*) téléchargé⁷, il convient de suivre la suite d'instructions présentée ci-après.

6. Actuellement, l'adresse donnée est la suivante : <http://mirrors.ircam.fr/pub/CTAN/systems/texlive/Images/texlive2014-20140525.iso>.

7. Il se trouve à l'adresse <http://www.ghostscript.com/download/gsdnld.html>.

Fenêtre	Action
Welcome	Next
License agreement	I agree
Choose install	Répertoire C:\Texlive\gs. Install
Completing	Finish

TABLE A.2 – Étapes d’installation de Ghostscript

GSview

GSview est un outil graphique qui facilite grandement l’utilisation de Ghostscript. Grâce à lui, la visualisation d’un document `.ps` est simplifiée.

La version de GSview à télécharger⁸ doit être compatible avec la version de Ghostscript. Une fois lancé l’exécutable, les étapes de configuration sont indiquées ci-après.

Fenêtre	Action
Winzip Self-Extractor	Setup
Select Language	Francais
This wizard will	Next
Copyright	Next
GSview can create	Next
Select a directory	Répertoire C:\Texlive\Ghostgum. Next
The directory you	Next
GSview setup will	Finish
Installation successful	Exit

TABLE A.3 – Étapes d’installation de GSview

Point important, il faut lancer une première fois GSview pour le configurer. Sans cela, il indiquera systématiquement une erreur à l’ouverture d’un fichier `.ps`. Une fois lancé, il faut aller dans le menu **Options** puis **Easy Configure** et sélectionner le numéro de version de Ghostscript⁹. GSview peut alors être fermé.

A.2.3 Configuration de Windows

Les différentes versions de Windows réagissent diversement à l’installation de \LaTeX . La plupart des versions¹⁰ ne feront pas fonctionner `ps2pdf`, mystérieusement introuvable. Il faut comprendre par là que lorsque Windows cherche un programme, il parcourt une liste de répertoires bien spécifiques et, dans notre cas, nos répertoires contenant \LaTeX ou Ghostscript n’en font pas partie.

Pour corriger ce point, il faut modifier la variable **Path** des variables d’environnement. Dans le panneau de configuration, double-cliquer sur **Système**¹¹. Sélectionner alors l’onglet avancé et cliquer alors sur **Variables d’environnement**.

8. À l’adresse <http://pages.cs.wisc.edu/~ghost/gsview/index.htm>, en retenant par rapport à la remarque ci-dessus la version **GSview release v5.0**.

9. Ce dernier est dans le nom du fichier d’installation de Ghostscript.

10. Windows 7 semble faire exception.

11. Dans le cas de Windows XP, il faut d’abord cliquer sur l’icône **Performance et maintenance**. Pour Windows Vista, il faut demander l’affichage « classique », l’icône **Système** apparaissant alors.

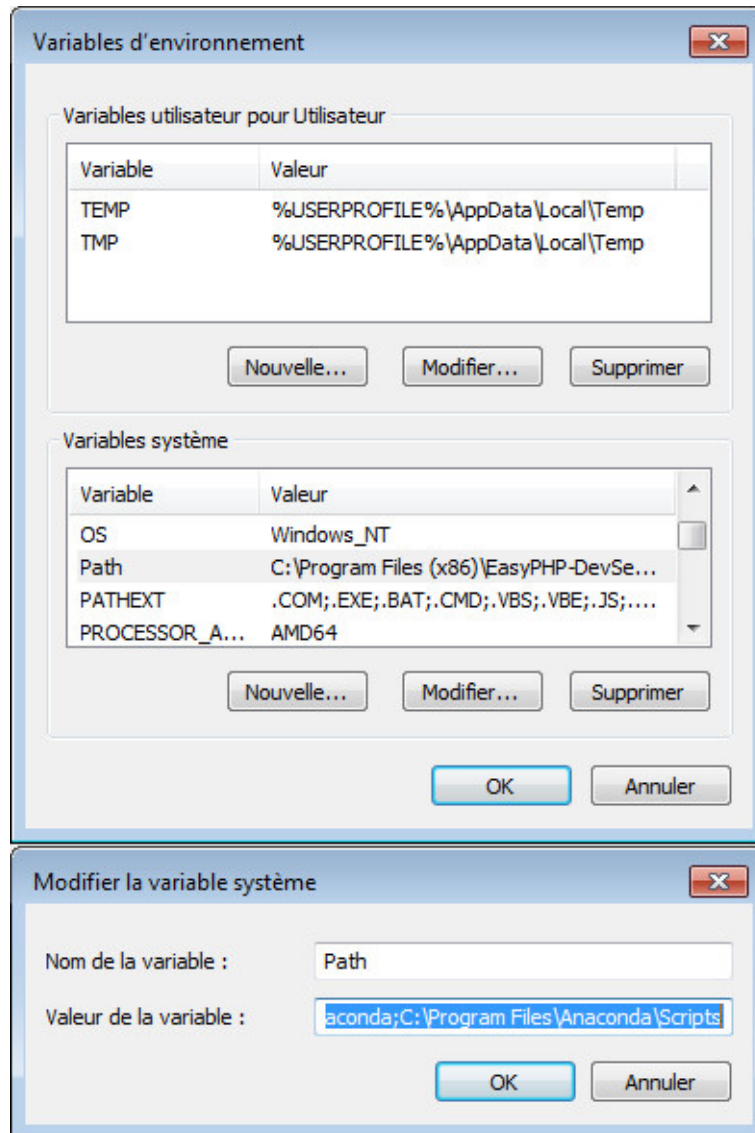


FIGURE A.2 – Exemple d'entrée des chemins sous Windows XP

Là, il faut sélectionner la ligne `Path` et cliquer sur `Modifier`. On ajoute alors¹² les chemins vers nos programmes, autrement dit la ligne suivante¹³ :

```
;C:\Texlive\bin\win32\ ;C:\Texlive\gs\gs9.06\bin\ ;C:\Texlive\gs\gs9.06\lib\
```

A.3 Des éditeurs

Une fois les différents programmes fondamentaux installés, il faut installer un programme permettant d'éditer de façon simple les documents `.tex` et de lancer aisément des compilations ou des conversions de document. Ce document en présente deux.

Dans tous les cas, un point d'attention est porté sur la configuration car il faut systématiquement indiquer à ces programmes où se trouvent les différents programmes que nous avons installé précédemment.

12. ... sans effacer les chemins déjà présents !

13. Les points-virgules servent ici de séparateurs entre les différents chemins.

A.3.1 Texmaker

Programme libre sous licence GNU, Texmaker¹⁴ est un des logiciels de référence pour l'édition de documents \LaTeX . L'installation se limite tout au mieux à indiquer le répertoire où sera installé le programme. Le paramétrage des programmes s'obtient par le menu **Options, Configurer Texmaker**. L'image ci-dessous¹⁵ donne la configuration attendue par défaut.

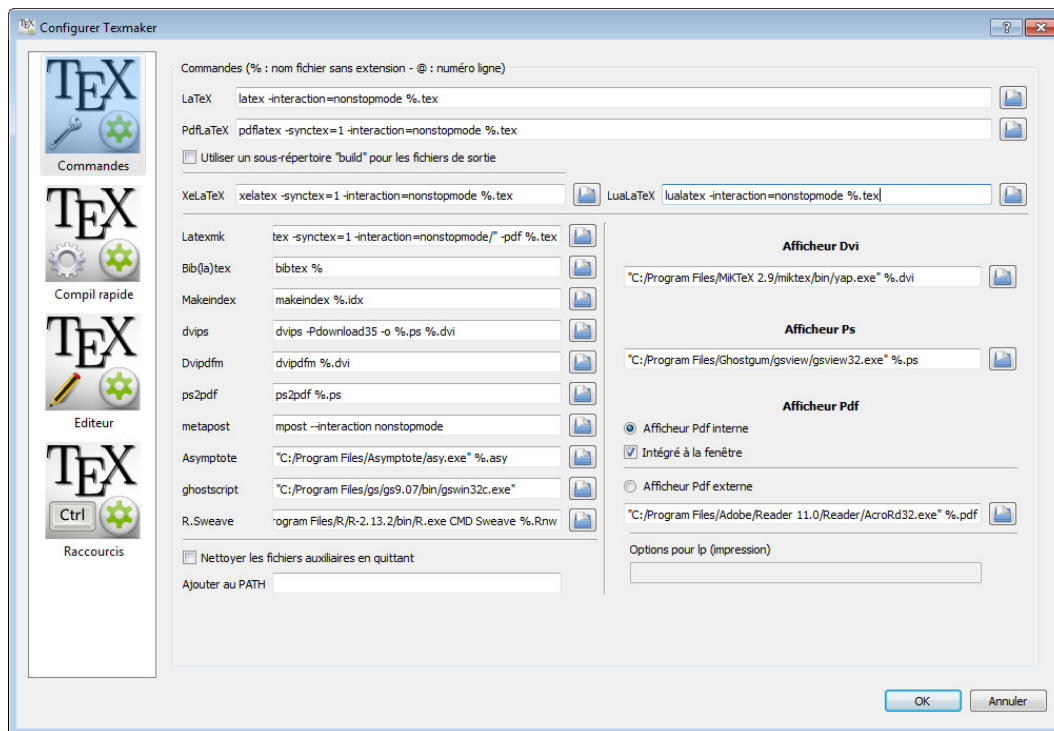


FIGURE A.3 – Paramétrage de Texmaker

Texmaker gère l'encodage UTF8. Il s'agit d'une norme de saisie des caractères qui se répand de plus en plus car elle permet de gérer de très nombreuses langues. Si vous utilisez UTF8, il faut modifier le préambule de notre document, en remplaçant `latin1` par `utf8`.

```
\usepackage[utf8]{inputenc}
```

CODE A.1 – Code pour l'UTF8

Selon la version de Texmaker, la gestion de l'UTF8 n'est pas systématique. Il faut parfois modifier directement l'encodage utilisé par défaut. Ceci se fait dans le menu **Options, Configurer Texmaker, Éditeur** en paramétrant dans l'encodage UTF-8.

A.3.2 TeXnicCenter

TeXnicCenter¹⁶ offre une large panoplie de fonctionnalités, à l'image de Texmaker. L'installation propose un peu plus de choix que dans Texmaker et demande également la localisation des fichiers de la distribution¹⁷. Les options proposées restent lisibles et rendent cette phase peu difficile.

14. Il est téléchargeable à l'adresse http://www.xm1math.net/texmaker/index_fr.html.

15. Cette présentation dépend de la version de Texmaker mais contient le même type d'information.

16. Téléchargeable à l'adresse <http://www.texniccenter.org/>.

17. Il faut indiquer le répertoire `C:\Texlive\2014\bin\win32` si vous avez suivi le paramétrage proposé dans ce document.

L'installation vous demandant de lui indiquer où est la distribution, il n'y a pas ou peu d'actions à faire pour configurer TeXnicCenter. À titre d'information, cette opération se fait dans le menu [Build](#) puis [Define Output Profiles](#).

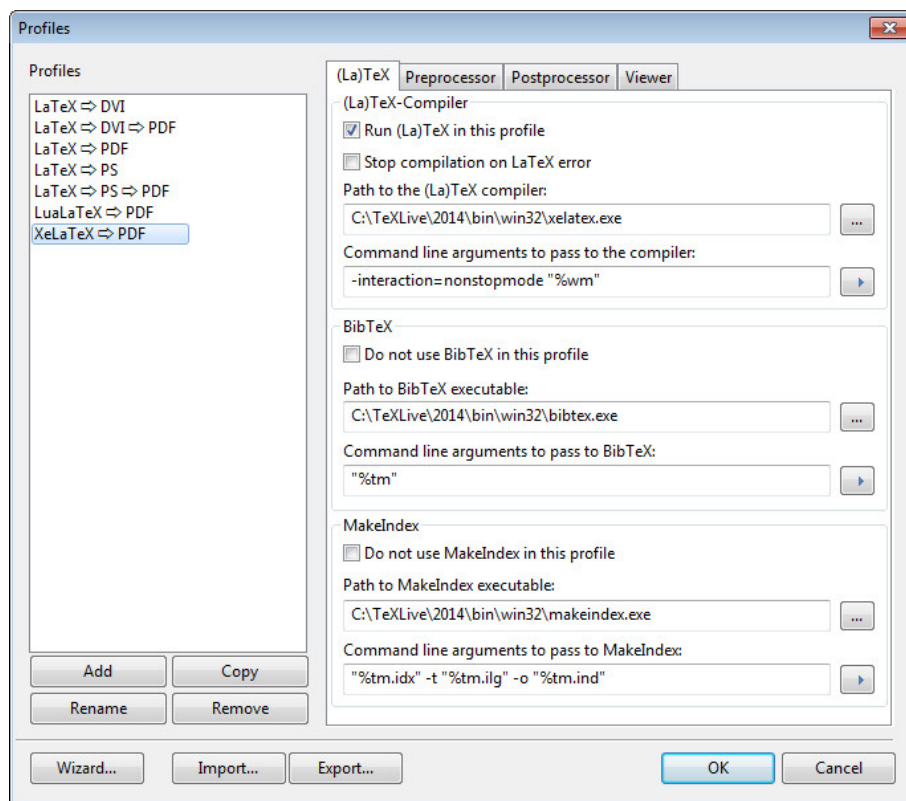


FIGURE A.4 – Paramétrage de TeXnicCenter

Annexe B

Liens avec d'autres programmes

B.1 Conversion d'image

Dans le cadre de la chaîne de compilation ps2pdf vue en page 10, L^AT_EX n'accepte que des images au format [eps](#). Pour convertir un fichier image en fichier [eps](#), de nombreux programmes sont disponibles. En voici un exemple avec le logiciel libre The Gimp¹.

Après avoir ouvert une image éditable avec The Gimp, il faut aller dans le menu [Fichier](#), [Exporter...](#) et le formulaire suivant apparaît.

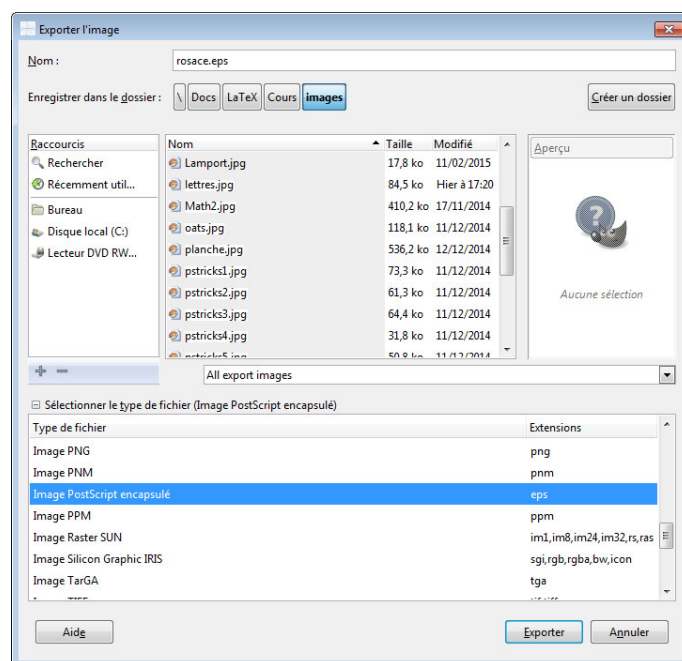
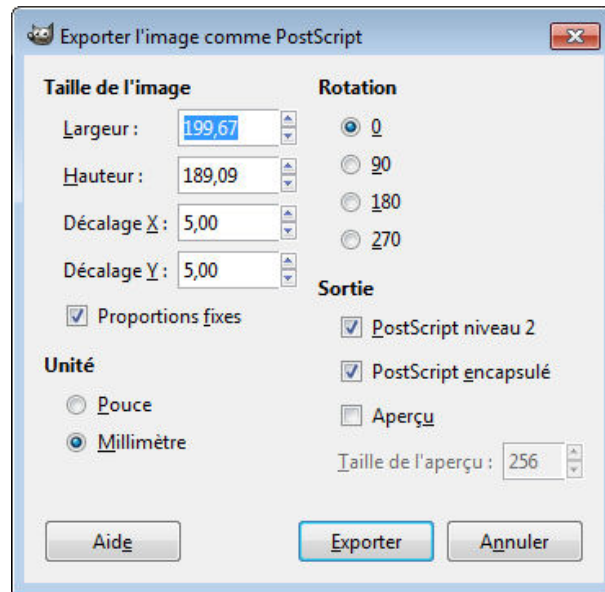


FIGURE B.1 – Formulaire d'exportation de The Gimp

Il suffit alors de changer le type de l'image en cliquant sur [Sélectionner le type de fichier](#) puis [Image postscript encapsulé](#) et enfin [Exporter](#). Dans la plupart des cas, les options proposées par défaut dans le second formulaire qui apparaît ne sont pas à modifier.

1. Il est téléchargeable à l'adresse <http://www.gimp.org/>.

FIGURE B.2 – Formulaire des options `.eps` de The Gimp

La figure s'insère alors dans \LaTeX en suivant la logique présentée en section 7.2.

B.2 Excel

Comme vu en 7.1.7 page 52, la macro complémentaire Excel2LaTeX permet de convertir une zone d'une feuille Excel en un code \LaTeX générant un tableau simple.

Mais la transmission d'information entre Excel et \LaTeX peut aller plus loin. \LaTeX travaille en effet sur des fichiers en format texte, format générable et manipulable par la plupart des langages de programmation, tel le Visual Basic. Voici donc un exemple de code Visual Basic permettant ceci :

```
Sub Test()
Open "C:\Test\liste.tex" For Output As #1
Avertissement = Cells(1, 1)
Signature = Cells(2, 1)
Print #1, "Attention :"
Print #1, "\begin{center}"
Print #1, Avertissement
Print #1, "\textsc{" & Signature & "}"
Print #1, "\end{center}"
Close #1
End Sub
```

CODE B.1 – Macro VB créant un code \LaTeX en encodage latin1

La procédure `Test`, avec la commande `Open For Output As #1`, va accéder en mémoire au fichier `liste.tex` dans le répertoire indiqué. Si le fichier n'existe pas, Excel le crée². Ce code récupère avec les deux lignes suivantes les valeurs présentes dans deux cellules de la feuille Excel active. Chaque ligne commençant par `Print #1`, insère alors la suite de la ligne dans le fichier `liste.tex`. La commande `Close #1` referme enfin le fichier `liste.tex`.

Les différentes lignes commençant par `Print #1`, insèrent des chaînes de caractère encadrées par le caractère `"` ou des variables. La concaténation de chaînes de caractère ou de variables se fait avec le caractère `&`³.

2. Mais si le répertoire n'existe pas, Excel ne le crée ici pas.

3. Ceci est vrai dans le code Visual Basic tout comme dans les formules apparaissant dans une feuille Excel.

Le fichier `liste.tex` peut alors être compilé avec \LaTeX s'il est complet ou être intégré dans un autre fichier avec les commandes `\include` ou `\input`, comme indiqué en section 8.1, page 58.

Pour travailler en UTF8, le code à retenir est plus complexe (mais aussi plus général) :

```
Sub Test()
' On teste l'existence du fichier final (pour le supprimer)
Set objFSO = CreateObject("Scripting.FileSystemObject")
If objFSO.FileExists("C:\Test\liste.tex") Then Kill ("C:\Test\liste.tex")

' Creation de l'objet contenant le texte LaTeX
Dim objStream As Object           ' Définition
Set objStream = CreateObject("ADODB.Stream") ' Création
objStream.Open                   ' Ouverture
objStream.Position = 0           ' On se place au début
objStream.Charset = "UTF-8"      ' Encodage UTF8

Avertissement = Cells(1, 1)
Signature = Cells(2, 1)
objStream.WriteText "Attention :" & vbCrLf
objStream.WriteText "\begin{center}" & vbCrLf
objStream.WriteText Avertissement & vbCrLf
objStream.WriteText "\textsc{" & Signature & "}" & vbCrLf
objStream.WriteText "\end{center}" & vbCrLf

' Sauvegarde finale et fermeture de l'objet créé
objStream.SaveToFile "C:\Test\liste.tex"
objStream.Close
End Sub
```

CODE B.2 – Macro VB créant un code \LaTeX en encodage UTF8

Ici, le `vbCrLf` permet d'insérer un retour à la ligne.

B.3 Les logiciels mathématiques

La précision de \LaTeX en matière de mathématiques en a fait un des outils les plus appréciés des mathématiciens. Aussi, de nombreuses applications mathématiques proposent la génération de codes ou de figures réutilisables immédiatement par \LaTeX .

B.3.1 R, avec Sweave

L'extension Sweave du logiciel R permet de créer un document `.tex` sur la base d'un document d'extension dédiée `.Snw` (ou `.Rnw`) placé dans le répertoire de travail usuel de R. Ce document contient du code \LaTeX classique mais peut contenir aussi du code R en respectant la norme suivante :

```
<< paramètres d'affichage >>=
Lignes de code R
@
```

CODE B.3 – Encadrement d'un code R sous \LaTeX

En voici quelques exemples appliqués :

```
<<>>=
library(stats)
x <- rnorm(20)
print(x)
print(t1 <- t.test(x))
@

<<echo=TRUE,print=TRUE>>=
1 + 1
1 + pi
sin(pi/2)
@
```

CODE B.4 – Exemples de code R sous \LaTeX

Sous R, ce document est exécutable avec Sweave en utilisant la série de commandes suivantes :

```
test <- system.file("Sweave", "fichier.Snw", package = "utils") \\
Sweave(test)
```

CODE B.5 – Instructions sous R

Une fois que R a effectué son traitement, il restitue un fichier `.tex` et autant d'images `.pdf` ou d'images `.eps`⁴ que de figures demandées à R. Le document \LaTeX doit contenir dans son préambule l'appel suivant :

```
\usepackage{Sweave}
```

CODE B.6 – Paquet nécessaire pour travailler avec R et Sweave

Le paquet `Sweave` se trouve dans les répertoires de R et peut être copié dans le même répertoire que votre fichier `.tex`. La suite revient à l'utilisation classique de \LaTeX .

Les possibilités sont ici nombreuses et, pour une vision bien plus complète, il existe un document de Jean Raymond LOBRY[23] sur la question.

4. Pour obtenir des images `.eps`, le fichier `.Snw` doit contenir l'instruction `\SweaveOpts{eps=true}`.

Annexe C

Un CV sous L^AT_EX

Pour obtenir un CV de qualité, le paquet qui sort actuellement du lot est `moderncv` de Xavier DANAUX. Ce paquet est documenté par des exemples et le cours ci-dessous reprend ce point pour permettre à tout à chacun d'y trouver de quoi faire. Le seul point non détaillé ici est la possibilité de faire le courrier avec le CV, le courrier étant proposé dans un format anglais.

C.1 Une nouvelle classe

Le paquet `moderncv` définit tout d'abord une nouvelle classe de document nommée `moderncv`. Cette dernière réagit de façon similaire aux options que sont `10pt`, `11pt`, `12pt`, `landscape`, `a4paper`, `a5paper` pour les plus courantes. Elle ajoute les deux options `roman` et `sans` qui permettent de rédiger respectivement en fonte romaine et en fonte linéale.

C.2 Différents thèmes

Ce paquet propose différents thèmes. Pour sélectionner son thème, il faut recourir à la commande `\moderncvtheme[couleur]{nom du thème}`. Le thème est à choisir parmi les valeurs suivantes : `banking`, `casual`, `classic` et `oldstyle`.

La couleur est à choisir par défaut parmi les valeurs : `black`, `blue`, `green`, `grey`, `orange`, `purple` et `red`.

Les couleurs peuvent éventuellement être redéfinies directement avec le bloc de commandes suivantes. Le bloc ci-dessous reprend le code pour obtenir l'option `blue` indiquée ci-dessus¹.

```
\definecolor{color0}{rgb}{0,0,0}           % noir
\definecolor{color1}{rgb}{0.22,0.45,0.70}  % bleu clair
\definecolor{color2}{rgb}{0.45,0.45,0.45}  % gris
```

CODE C.1 – Définition de couleur pour le CV

La couleur `color0` joue sur le texte, `color1` sur les règles et les titres de section et `color2` sur les coordonnées, l'encadrement de la photo et le nom.

C.3 Paramètres personnels

Différentes commandes placées dans le préambule du document sont créées pour stocker les coordonnées obligatoires de l'auteur du CV :

1. Ce bloc correspond à ce que font les différents appels d'options et qui se retrouvent dans les fichiers `moderncvcolor...sty`.

- `\firstname{prénom}` définit le *prénom* ;
 - `\familyname{nom}` définit le *nom*.
- Des commandes optionnelles sont également disponibles :
- `\title{titre}` définit le *titre* du CV ;
 - `\address{adresse}{code postal, ville}` définit l'adresse en deux champs ;
 - `\phone{numéro}` définit le *numéro* de téléphone fixe ;
 - `\mobile{numéro}` définit le *numéro* de téléphone mobile ;
 - `\fax{numéro}` définit le *numéro* de fax ;
 - `\email{mail}` définit l'adresse *mail* ;
 - `\homepage{page web}` définit l'adresse d'une *page web* ;
 - `\extrainfo{infos}` définit des *données* complémentaires ;
 - `\quote{citation}` définit une citation ;
 - `\photo[dimension] [marge] {image}` définit une photo de hauteur *dimension* avec une marge d'épaisseur *marge*. Il doit s'agir du nom de l'*image*, `.eps` ou `.jpg` selon la chaîne de compilation retenue, sans indiquer l'extension.

C.4 Découpage du CV

La partie en-tête (prénom, nom) et la partie pied de page sont générées automatiquement avec la commande `\makecvttitle` placée dans le corps du document. Le découpage du CV reste par contre à la main du rédacteur. La découpe du CV en différents thématiques se fait par les commandes `\section{}` et `\subsection{}`. Les différentes lignes du CV peuvent être formatées spécifiquement de quatre façons :

- `\cvitem{texte court}{texte long}`² met l'argument *court* en valeur (gras ou en colonne), l'argument long étant mis à côté pour donner des détails ;
- `\cvdoubleitem{court}{long}{court}{long}`³ reprend l'idée de `\cvline` mais en traitant deux entrées en même temps par un jeu de double colonne ;
- `\cvitemwithcomment{court}{long}{droite}`⁴ rajoute un élément placé sur la même ligne à *droite* des deux premiers ;
- `\cventry{dates}{court}{société}{ville}{pays}{long}` structure encore plus l'information. Elle est à réserver pour le détail des expériences ou des formations.

En parallèle, des commandes d'énumération sont proposées par ce paquet :

- `\cvlistitem{donnée}` ;
- `\cvlistdoubleitem{donnée 1}{donnée 2}` propose la même énumération mais sur deux colonnes.

C.5 Commandes complémentaires

L'observation de plusieurs exemples d'utilisation de `moderncv` montre qu'il est souvent fait appel aux deux commandes suivantes dans le préambule du document pour affiner la présentation du CV :

- `\usepackage[scale=0.8]{geometry}` afin de régler les dimensions du CV par rapport à la feuille de papier par l'appel de `geometry` ;
- `\nopagenumbers{}` qui empêche la numérotation des pages du CV.

2. Elle est aussi nommée `\cvline`.

3. Elle est aussi nommée `\cvcomputer`.

4. Elle est aussi nommée `\cvlanguage`.

C.6 Exemple

```
\documentclass[10pt,a4paper]{moderncv}
\moderncvtheme[blue]{oldstyle}
\usepackage[scale=0.8]{geometry}
\firstname{Garfield}
\familyname{Lechat}
\title{Chat domestique}
\address{8 rue Lasagne}{Cheshire}
\photo[64pt]{garfield}

\begin{document}
\makecvtitle

\section{Expériences}
\cventry{2010}{Testeur de matelas}{Dodo}{Le Mans}{France}{}
\cventry{2008--2009}{Coinceur de bulle}{Sommeil blanc}{Poitiers}{France}{}

\end{document}
```

CODE C.2 – Exemple de CV

C.7 Pour aller plus loin

Parmi les autres modèles de CV qui peuvent être trouvés sur l'Internet, citons ici :

- celui d'Olivier BICHLER⁵ utilisant une classe permettant de faire un CV avec contenu optionnel;
- celui d'Adrien FRIGGERI⁶, très graphique, utilisant une classe se basant en particulier sur les fonctionnalités de \LaTeX et *tikz*.
- le site *LaTeX templates*[2] propose également de nombreuses autres maquettes toutes aussi intéressantes⁷.

5. <http://olivier.bichler.free.fr/www/?page=content/index&blog_content_articleId=26>.

6. <<https://github.com/afriggeri/CV>>.

7. <<http://www.latextemplates.com/cat/curricula-vitae>>.

Annexe D

La page de garde

D.1 Définition et exemple

L'extension `pagedegarde` permet d'inclure dans un document \LaTeX la page de garde « officielle » de l'Institut des Actuaire déclinée avec les spécificités de chaque filière, telle qu'elle apparaît ci-après sur un exemple pour l'ISUP.

Voici un exemple de code permettant de la générer. Les appels d'extensions usuelles de début de document n'y sont pas indiquées.

```
\documentclass[a4paper,11pt,twoside]{article}
\usepackage[frenchb]{babel}
\usepackage[T1]{fontenc}
\usepackage[latin1]{inputenc}
\usepackage{lmodern}

\usepackage[confiun,hyperref,isup]{pagedegarde}

\dategarde{22 février 1963}
\auteurgarde{Jean}{Dieudonné}
\titredegarde{Tentative de définition systématique d'une (...)}
\juryiagarde{Nicolas}{Bourbaki}
\juryyagarde{Carl Friedrich}{Gauss}
\jurybgarde{David}{Hilbert}
\directeurgarde{Pierre-Simon}{de Laplace}
\invitegarde{Jean Baptiste Joseph}{Fourier}

\begin{document}
\pagedegarde
  Le reste du document...
\end{document}
```

CODE D.1 – Exemple de génération de la page de garde

Ce code est fourni dans le fichier `testdegarde.tex` qui accompagne cette documentation, le fichier d'extension `pagedegarde.sty` et les images dans leur répertoire dédié (répertoire dont le nom ne doit pas être modifié).

**Mémoire présenté le 22 février 1963
pour l'obtention du diplôme
de Statisticien Mention Actuariat
et l'admission à l'Institut des Actuaire**

Par : Jean DIEUDONNÉ

Titre : Tentative de définition systématique d'une nomenclature exhaustive des mémoires d'actuariat à travers les âges

Confidentialité : oui, un an. *Les signataires s'engagent à respecter la confidentialité indiquée.*

***Membre présent du jury de l'Institut
des Actuaire***

Nicolas BOURBAKI

Signature :

Entreprise

Signature :

Membres présents du jury de l'EURIA

Carl Friedrich GAUSS

David HILBERT

Directeur de mémoire en entreprise

Pierre-Simon DE LAPLACE

Signature :

Invité

Jean Baptiste Joseph FOURIER

Signature :

***Autorisation de publication et de mise en ligne sur un site de diffusion
de documents actuariels
(après expiration du délai de confidentialité)***

Signature du responsable entreprise

Signature du candidat

Bibliothèque :

Secrétariat :

D.2 Appel de l'extension

L'extension `pagedegarde` est munie de plusieurs options permettant d'effectuer des réglages de la présentation de la page de garde. L'appel avec option se fait sous la forme suivante :

```
\usepackage[option1,option2,...]{pagedegarde}
```

Il est recommandé de charger cette extension plutôt en fin de préambule du document. En effet, elle charge par défaut les extensions suivantes : `ifpdf`, `xcolor` (avec son option `dvipsnames`), `graphicx`, `xspace` et `geometry`. Elle peut également, par le biais des options, charger l'extension `hyperref`, cette dernière devant normalement être la dernière chargée.

D.2.1 L'option de filière

Par défaut, le document est considéré comme un mémoire sans filière : seul le logo de l'IA apparaît et il est évoqué le terme de « filière ». Un choix d'option doit être effectué pour modifier les affichages par défaut.

Trois filières sont définies :

- l'EURIA avec l'option `euria`;
- l'EURIA en lien avec Télécom Bretagne avec l'option `telecomb`;
- l'ISUP avec l'option `isup`.

D.2.2 L'option de brouillon

Par défaut, le document est présenté en mode finalisé. Un mode brouillon est cependant disponible par le biais de l'option `draft`. Il permet d'observer les différents champs paramétrables en gris. Si ces champs n'ont pas valeur, `pagedegarde` affiche alors la commande permettant de remplir ce champ à l'endroit où il sera affiché.

D.2.3 Les options de confidentialité

Par défaut, le document est considéré comme non confidentiel. Pour spécifier une durée de confidentialité, cinq options sont disponibles selon la durée de confidentialité :

- `confiun` : un an ;
- `confideux` : deux ans ;
- `confitrois` : trois ans ;
- `confiquatre` : quatre ans ;
- `conficinq` : cinq ans.

D.2.4 L'option de page verso uniquement

Par défaut, le verso de la page de garde reste totalement vide, présentation usuelle pour un document en recto-verso. Ce comportement peut être modifié en spécifiant l'option `versoseul`. La page suivant immédiatement la page de garde est alors accessible.

D.2.5 L'option de liens hypertextes

L'option `hyperref` charge l'extension `hyperref` gérant les liens hypertextes dans les documents PDF. Sur la page de garde, les sigles ainsi que les noms des filières et autres organismes renvoient alors vers les sites internet associés.

L'utilisation de cette option sélectionne certains réglages de l'extension `hyperref` et ceci peut influencer sur la présentation de l'ensemble du document. Il faut donc ici spécifier après le chargement de l'extension `pagedegarde` les options nécessaires à l'extension `hyperref` pour présenter le reste du mémoire. Ceci s'obtient avec la commande `\hypersetup{options}`, les *options* étant celles qui se retrouvent pour l'appel du paquet `hyperref`.

D.3 Paramètres

Les champs paramétrables de la page de garde sont alimentés par différentes commandes. La liste en est la suivante :

Commande	Affichage
<code>\dategarde{date}</code>	Date de la présentation du mémoire
<code>\auteurgarde{prénom}{nom}</code>	Identité de l'auteur du mémoire
<code>\titregarde{titre}</code>	Titre du mémoire
<code>\juryiagarde{prénom}{nom}</code>	Identité du jury IA
<code>\juryagarde{prénom}{nom}</code>	Identité du 1 ^{er} jury filière
<code>\jurybgarde{prénom}{nom}</code>	Identité du 2 ^e jury filière
<code>\jurycgarde{prénom}{nom}</code>	Identité du 3 ^e jury filière
<code>\jurydgarde{prénom}{nom}</code>	Identité du 4 ^e jury filière
<code>\juryegarde{prénom}{nom}</code>	Identité du 5 ^e jury filière
<code>\juryfgarde{prénom}{nom}</code>	Identité du 6 ^e jury filière
<code>\directeurgarde{prénom}{nom}</code>	Identité du directeur de mémoire
<code>\invitegarde{prénom}{nom}</code>	Identité de l'invité
<code>\entreprisegarde{entreprise}</code>	Entreprise associée au mémoire

TABLE D.1 – Les paramètres de `pagedegarde`

D.4 Affichage

L'affichage de la page de garde est obtenu en plaçant la commande `\pagedegarde` à l'endroit souhaité dans le corps du document. Aussi, cette page pourrait donc être placée n'importe où dans le document.

Dans l'exemple de code présenté au début de ce document, la position de la commande correspond à la position la plus naturelle : elle produit la page de garde en toute première page.

D.5 Commandes complémentaires

L'extension `pagedegarde` crée quelques commandes spécifiques aux organismes cités dans la page. Ces commandes sont ainsi toutes réutilisables.

D.5.1 Commandes de texte

Cinq commandes écrivant du texte sont mises à disposition :

Commande	Affichage
<code>\EURIA</code>	EURIA
<code>\Euria</code>	Euria
<code>\euria</code>	EUro Institut d'Actuariat
<code>\euriamail</code>	euria@univ-brest.fr
<code>\ia</code>	Institut des Actuaire
<code>\ISUP</code>	ISUP

TABLE D.2 – Commandes complémentaires de texte de `pagedegarde`

D.5.2 Commandes de sigle

Quelques commandes sont mises à disposition pour afficher des sigles hors de la page de garde. Tous les sigles sont conditionnés à l'affichage par une dimension.

Commande	Affichage
<code>\euriasigle{dimension}</code>	Sigle de l'EURIA
<code>\iasigle{dimension}</code>	Sigle de l'IA
<code>\isupsigle{dimension}</code>	Sigle de l'ISUP
<code>\telbsigle{dimension}</code>	Sigle de Télécom Bretagne
<code>\ubosigle{dimension}</code>	Sigle de l'UBO

TABLE D.3 – Commandes complémentaires de sigle de `pagedegarde`

Voici les différents sigles :



FIGURE D.1 – Sigles

Le sigle de l'EURIA est composé d'un texte et d'une image. Le texte peut donc être affiché dans des polices de caractères différentes, la ligne colorée s'ajustant à la longueur du texte.

D.5.3 Commandes avec lien hypertexte

Si l'option `hyperref` de `pagedegarde` est sélectionnée, il suffit de rajouter le suffixe `ref` aux commandes vues dans cette section D.5 pour obtenir des commandes générant un lien hypertexte dans le document PDF. Par exemple :

- `\EURIA` devient `\EURIAref` ;
- `\ubosigle{dimension}` devient `\ubosigleref{dimension}` et ainsi de suite.

En l'absence de cette option, ces commandes deviennent strictement équivalentes aux commandes sans le suffixe `ref`.

Annexe E

Avant de réduire votre ordinateur en poussière

E.1 Quelques réflexes de survie

E.1.1 Réflexes pour amateurs

\LaTeX indique normalement le numéro de ligne où se situe l'anomalie qui l'empêche de compiler le document. Toutefois, ce numéro de ligne peut parfois être faux ! La recherche des erreurs peut alors se faire en déplaçant dans le document la commande `\end{document}` puis compiler. Ceci permet de déterminer, après quelques essais, quelle partie du code ou quelle ligne de code crée l'anomalie de compilation.

Certaines erreurs peuvent également survenir à retardement car elles peuvent être transmises aux fichiers réutilisés par \LaTeX lors des compilations suivantes comme `\aux`, `\ind`... Ce cas est souvent délicat à analyser. Aussi, si une erreur se produit sans raison apparente, la suppression des fichiers générés par la compilation peut parfois aider à résoudre le problème¹.

E.1.2 Réflexes pour passionnés

\LaTeX dispose de primitives lui permettant de décrire de façon beaucoup plus détaillée les travaux qu'il exécute :

- `\tracingcommands=1` (ou 2) affiche le détail des commandes utilisées, le chiffre précisant la profondeur de l'analyse effectuée ;
- `\tracingall` affiche beaucoup plus d'informations que la première commande en restituant dans le fichier journal le contenu des pages, les définitions et redéfinitions que \LaTeX opère.

Ces analyses demandent une connaissance de \LaTeX assez avancée. Les lecteurs intéressés pourront lire à profit la très complète annexe B de [16]².

E.2 Le bestiaire monstrueux

La liste qui suit présente quelques erreurs fréquemment rencontrées lors du cours. Cette liste n'est en aucun cas exhaustive : pour analyser d'autres erreurs, l'annexe B de [16] citée à la section précédente devrait pouvoir apporter un complément d'information appréciable.

1. Il s'agit souvent de la dernière opération de correction d'une erreur... qui, si elle échoue, conduit à donner des coups de tête au premier mur qui passe.

2. Disponible en ligne : <http://latex-project.org/guides/lc2fr-apb.pdf>.

Bad math environment delimiter

Le cas survient lorsqu'un environnement mathématique a été ouvert avec la balise fermante, par exemple en écrivant `\]` au lieu de `\[`.

Display math should end with \$\$

La fermeture d'un environnement mathématique hors ligne n'a pas été faite avec la bonne commande, par exemple en mettant un `$` au lieu d'un `\]` après avoir mis un `\[`. Il est à noter que le `$$`, pourtant suggéré ici par \TeX , doit être évité.

Extra }, or forgotten \$

Le cas survient lorsqu'un environnement mathématique a été ouvert sans être refermé.

File ended while scanning use of...

Cette erreur indique qu'une accolade ouvrante n'a jamais été suivie de l'accolade fermante (\TeX allant la chercher jusqu'à la fin du fichier). Cette erreur est suivie du nom de la commande pour laquelle il manque l'accolade fermante.

Missing \$ inserted

Les commandes présentant des mathématiques requièrent un environnement mathématique. Ceci apparaît le plus souvent en présence de caractères dédiés comme `_`, `^`, de lettres grecques comme `\alpha` ou de commandes mathématiques comme `\sum`, ceci sans que le mode mathématique ait été appelé (avec par exemple `$` ou `\[`).

No room for a new...

Le chargement de nombreux paquets peut saturer la mémoire allouée (des registres) à \TeX ³. Pour contourner cette difficulté, il peut être fait usage des commandes suivantes en début de préambule, le `28` étant un paramètre libre :

```
\usepackage{etex}
\reserveinserts{28}
```

CODE E.1 – Augmentation de capacité pour \TeX

Too many }'s

Fréquente en cas de formules mathématiques, cette erreur a pour source un caractère `}` en trop ou un caractère `{` manquant.

Undefined control sequence

Normalement suivie de la commande coupable, vous vous trouvez ici face à une commande dont \TeX n'a pas la définition. Trois possibilités :

- vous n'avez pas chargé le paquet définissant cette commande ;
- vous n'avez pas défini cette commande dans le préambule ;
- vous ne savez pas utiliser vos doigts⁴.

3. Voir <http://www.tex.ac.uk/cgi-bin/texfaq2html?label=noroom>.

4. Vous connaissant, ce cas s'avère être le plus probable.

Annexe F

Index des paquets et programmes

A

algorithm2e, 29
amssymb, 40
array, 50
arydshln, 53

B

babel, 14, 17, 63
beamer, 14, 113
biblatex, 36
bookman, 15

C

calc, 63
caption, 69
chancery, 15
charter, 15
colortbl, 51
comment, 63
concrete, 15

D

dvips, 9, 10

E

enumitem, 69
euler, 15
eurosym, 6, 17
Excel, 52, 96
Excel2LaTeX, 52, 96

F

fancybox, 22
fancyhdr, 36, 66
floatrow, 32, 68, 69
fncychap, 69
fontenc, 12
FontForge, 82
fontspec, 4, 12, 82

fourier, 15

fouriernc, 15

framed, 22

G

geometry, 66, 100, 104
Ghostscript, 88–91
glossaries, 34
graphicx, 53, 55, 70, 104
GSview, 88–91

H

hhline, 50
histogra, 75
hyperref, 13, 23, 28, 104

I

ifpdf, 63, 104
ifthen, 62
index, 34
inputenc, 12

K

kerkis, 15
kmath, 15

L

latex, 10
latexdiff, 28
layout, 64
layouts, 64
lettre, 13
lettrine, 29
lifecon, 46
listings, 29
lmodern, 14
longtable, 53
lualatex, 10

M

makeidx, 33
makeindex, 33, 34, 68
mathptmx, 15
mathtools, 40
memoir, 13
minitoc, ii, 9, 28, 32
moderncv, 14, 99, 100
multicol, 29, 67
multido, 62, 76
multirow, 50

N

newcent, 15

P

pagedegarde, 37, 102, 104–106
palatino, 15
pdflatex, 10, 12, 53
picins, 56
pifont, 18, 19, 27
ps2pdf, 9, 10, 53, 90, 91, 95
pst-plot, 73, 74, 76
pstricks, 4, 10, 23, 70–73, 75–79, 113

R

R, 97, 98, 113

S

scrartcl, 13
scrlettr2, 13
scrbook, 13
scrreprt, 13
smartref, 33
stmaryrd, 40
Sweave, 98
Sweave, 97, 98

T

texindy, 34

Texmaker, 12, 34, 88, 93

TeXnicCenter, 93, 94

textcomp, 18

The Gimp, 95, 96

tikz, 70, 77–79, 101, 112, 114

titlesec, 69

tocbibind, 32

tocloft, 32

todonotes, 28

U

ulem, 21

V

varioref, 33

Visual Basic, 96

X

xcolor, 23, 24, 85, 104

xelatex, 10

xetex, 10

xkeyval, 63

xspace, 60, 85, 104

xwatermark, 29

Annexe G

Index des concepts

B

bascule, 7
bibliographie, 34

C

casseau, 18
catcode, 6
chasse fixe, 16
compilation, 8
corps, 5, 13
couleur, 23, 28, 51

D

délimiteur, 6
dingbat, 18
distribution, 89
division, 31, 63

E

empattements, 16
en-tête, 66
encodage, 4
environnement, 8
erreur, 107
extension, 3

F

famille, 16
fichier
 ◦aux, 9, 10, 107
 ◦bbl, 36
 ◦bib, 35, 36
 ◦cls, 85
 ◦dvi, 8–10
 ◦eps, 53, 89, 90, 95, 96, 98, 100

◦idx, 33, 34
◦ind, 34, 107
◦ist, 34, 68
◦jpg, 10, 53, 77, 100
◦lof, 28
◦log, 9, 10, 20, 33
◦lot, 28
◦mtc, 9, 28
◦otf, 81
◦pdf, 8–10, 12, 14, 28, 29, 77, 90, 98
◦ps, 9, 10, 89–91
◦Rnw, 97
◦Snw, 97, 98
◦sty, 83
◦tex, 5, 8, 10, 12, 19, 27, 30, 34–36, 46, 48, 53, 58, 68, 92, 93, 97, 98
◦toc, 28, 31
◦ttf, 81

figure, 53
flottant, 47, 68
format, 3
forme, 16

G

groupe, 7

I

image, 53, 56, 95
indentation, 19
index, 33, 67
installation, 88

J

journal, 9

M

macrocommande, 3
mathématiques, 38
mode mathématique, 38

N

notes, 27

P

page, 64
 dimensions, 64
paquet, 3
pied de page, 66
police de caractère, 15
polices de caractères, 81
préambule, 5

R

référence, 33

S

source, 5
structure, 30
série, 16

T

table des matières, 31
 correction, 32
 par subdivision, 32
tableau, 47
T_EX
 histoire, 2
T_EXLive, 89
titre, 36

Bibliographie

- [1] *L^AT_EX Stack Exchange*.
<<http://tex.stackexchange.com/>>.
Probablement le meilleur forum anglophone pour poser des questions sur T_EX et ses descendants, d'autant que de très grands experts viennent apporter leur aide.
- [2] *L^AT_EX Templates*.
<<http://www.latextemplates.com/>>.
Pour apporter un peu de variété à un document L^AT_EX, ce site propose différents modèles de présentation avec le code L^AT_EX associé.
- [3] *T_EXample.net*.
<<http://www.texample.net/>>.
Ce site regroupe de nombreux exemples d'utilisation de *tikz*, toujours avec le code associé.
- [4] *Lexique des règles typographiques en usage à l'Imprimerie nationale*. Imprimerie nationale, 2002.
Des abréviations à la zoologie en passant par les mathématiques, ce petit livre fait le tour de ce qu'il faut savoir pour présenter un texte francophone.
- [5] Jacques ANDRÉ : *Petites leçons de typographie*. 2012.
<<http://jacques-andre.fr/faqtypo/lessons.pdf>>.
Vous ne lirez et ne composerez plus un texte de la même manière après avoir lu ces leçons d'un auteur réputé dans le monde de la composition numérique.
- [6] Walter APPEL, Céline CHEVALIER, Emmanuel CORNET, Sébastien DESREUX, Jean-Julien FLECK et Paul PICHAUREAU : *L^AT_EX pour l' impatient*. H&K, 2009.
D'un format un peu inhabituel, cet ouvrage traite de très nombreux aspects de L^AT_EX avec le recul d'utilisateurs chevronnés (et s'avère être l'un des moins chers).
- [7] Iana ATANASSOVA : *Mon blog informatique — Astuces et actualités en L^AT_EX*.
<<http://monbloginformatique.blogspot.fr/>>.
Un blog actif (et esthétique) en français sur L^AT_EX.
- [8] Benjamin BAYART *et al.* : *FAQ de fr.comp.text.tex*. 2004.
<<http://www.grappa.univ-lille3.fr/FAQ-LaTeX/>>.
Si quelqu'un pense avoir des questions nouvelles sur L^AT_EX, qu'il jette un coup d'œil à cette Foire Aux Questions, document de 370 pages regroupant les questions et l'expertise du groupe de discussion `fr.comp.text.tex`.
- [9] Denis BITOUZÉ et Jean-Côme CHARPENTIER : *L^AT_EX, l'essentiel*. Pearson Education France, 2010.
De par sa publication récente, cet ouvrage donne la présentation la plus à jour et la plus intuitive de L^AT_EX, présentant ainsi certains des derniers paquets les plus pratiques. Il évoque également, chose plus rare, les glossaires et les présentations projetées.
- [10] Jean-Pierre COLIGNON : *Un point, c'est tout !* Victoires Éditions, 2011.
Pas moins de 152 pages pour un sujet qui aurait pu sembler intuitif...mais qui s'avère être riche de surprises et de sens ! Jean-Pierre COLIGNON, qui a été chef correcteur du *Monde*, présente ici l'étendue de ses connaissances sur ce sujet inattendu.

- [11] Sébastien COMBÉFIS : *L^AT_EX HowTo*. 2010.
<http://www.latex-howto.be/files/LaTeX-HowTo-full.pdf>.
 Un large éventail des possibilités de L^AT_EX avec de très nombreux exemples à l'appui.
- [12] Fabien CONUS et Franck PASTOR : *Introduction à L^AT_EX sous MAC OS X*. 2012.
<http://redac.cuk.ch/franck/article-cuk-latex-pdf/intro-latex-rev-bis.pdf>.
 Pour les détenteurs de Mac, voici un document présentant l'installation et l'utilisation de L^AT_EX sous Mac, cas particulier non détaillé dans ce cours.
- [13] Bernard DESGRAUPES : *L^AT_EX. Apprentissage, guide et référence*. Vuibert, 1999.
 Un des meilleurs ouvrages pour qui veut rentrer plus avant dans le fonctionnement de L^AT_EX. Le chapitre sur la personnalisation reste un des plus intéressants. Il ne lui manque que la présentation des graphiques qui, selon l'auteur, devrait faire l'objet d'un ouvrage à part entière.
- [14] Arnaud GAZAGNES : *L^AT_EX... pour le prof de maths*. 2015.
<http://math.univ-lyon1.fr/irem/spip.php?article340>.
 Un cours très complet sur l'utilisation de L^AT_EX. Destiné aux enseignants, il donne également beaucoup d'éléments pour toute personne intéressée par le sujet, présentant en particulier de façon détaillée *beamer* et *pstricks*.
- [15] Michel GOOSSENS, Frank MITTELBAACH, Sebastian RAHTZ, Denis ROEGEL et Herbert VOSS : *The L^AT_EX Graphics Companion*. Addison-Wesley, 2010.
 Sans doute le seul ouvrage entièrement dédié l'ensemble des manières de générer des graphiques avec L^AT_EX : par l'importation d'images ou par la génération des graphiques avec L^AT_EX, Metapost ou *pstricks*. Une mine pour les amateurs de graphiques.
- [16] Michel GOOSSENS, Frank MITTELBAACH et Alexander SAMARIN : *L^AT_EX Companion*. Campus-Press, 1996.
<http://latex-project.org/guides/lc2fr-apb.pdf>.
 Ouvrage « fleuve » détaillant très précisément les paquets majeurs accompagnant L^AT_EX (d'où son volume). Il est à réserver aux spécialistes ou aux personnes cherchant des fonctionnalités très spécifiques. Le lien cité ici présente la seule annexe B sur la gestion des erreurs.
- [17] Philippe GOUTET : *Module LM204 — Initiation à L^AT_EX*. 2011.
<http://pgoutet.free.fr/latex/>.
 Ce cours, dans la prolongation directe de celui de Manuel PÉGOURIÉ-GONNARD[30], propose de nombreux exercices et développements avancés. Il contient également un aide mémoire bien pensé.
- [18] Jean-Michel HUFFLEN, Denis ROEGEL et Karl TOMBRE : *Guide local (L^A)T_EX*. 1998.
<http://tex.loria.fr/guide/guide1998.pdf>.
 Un classique de la documentation sur L^AT_EX, contenant en particulier des éléments sur la gestion des polices de caractères et un ensemble de règles typographiques.
- [19] Donald E. KNUTH : *The T_EXbook*. Addison-Wesley, 1986.
 Le T_EXbook, écrit par KNUTH lui-même, est le point de départ de ceux qui souhaitent comprendre la mécanique interne de T_EX. Entre descriptions et exercices à la difficulté progressive, l'auteur y glisse même de l'humour !
- [20] Thierry LACHAND-ROBERT : *La maîtrise de T_EX et L^AT_EX*. Masson, 1995.
 Très orienté sur le code T_EX, cet ouvrage sera précieux pour qui veut développer de nouvelles fonctionnalités.
- [21] Leslie LAMPORT : *L^AT_EX – A Document Preparation System – User's Guide and Reference Manual*. Addison-Wesley, 1994.
 L'ouvrage signé de l'auteur de L^AT_EX. Pas forcément le plus simple et le plus complet, il reste cependant une très bonne référence en anglais.
- [22] Jeanne LEGRAND et Tanguy ORTOLO : *L^AT_EX à 200%*. O'Reilly, 2007.
 Si optimiser une procédure vous intéresse, ce livre présente une série de *hacks* qui pourraient bien vous simplifier la vie tel « réaliser des compilations côté serveur » ou « réaliser une affiche ».
- [23] Jean Raymond LOBRY : *Comment rédiger un rapport avec la commande Sweave() de R*. 2009.
<http://pbil.univ-lyon1.fr/R/pdf/tdr78.pdf>.
 Une explication très complète pour interfacer R et L^AT_EX.

- [24] Vincent LOZANO : *Tout ce que vous avez toujours voulu savoir sur \LaTeX sans jamais oser le demander*. In Libro Veritas, 2008.
<<http://framabook.org/246/>>.
Un très bon document pour qui voudrait comprendre la méthode pour modifier le comportement de \LaTeX . D'autant que le livre autocommente amplement son propre style ! Cet ouvrage est libre et gratuit (l'édition papier pouvant être achetée pour soutenir cette excellente initiative).
- [25] Bertrand MASSON : *Les fiches à Bébert*.
<<http://lesfichesabebert.fr/>>.
Un wiki avec des fiches souvent très détaillées.
- [26] Tobias OETIKER, Hubert PARTL, Irene HYNÄ et Elisabeth SCHLEGL : *Une courte (?) introduction à \LaTeX 2_ε*. 2011.
<<http://www.ctan.org/tex-archive/info/lshort/french/>>.
Un autre classique de la documentation de qualité et libre sur \LaTeX , ce document évoluant depuis plus de 10 ans.
- [27] Scott PAKIN : *The comprehensive \LaTeX symbol list*. 2008.
<<http://www.ctan.org/tex-archive/info/symbols/comprehensive/>>.
Environ 5913 symboles à portée de clic sur le site du CTAN !
- [28] Franck PASTOR : *Articles sur CUK.fr*.
<<http://www.cuk.ch/articles/author/franck-pastor>>.
Une série d'articles suivants les récentes évolutions concernant \LaTeX .
- [29] Yves PERROUSSEAU : *Manuel de typographie française élémentaire*. Atelier Perrousseau Éditeurs, 1997.
Le typographie est un art, une histoire et une technique. Yves PERROUSSEAU, typographe et éditeur, en fait un état savant et pratique.
- [30] Manuel PÉGOURIÉ-GONNARD : *Formations \LaTeX au CIES Jussieu*. 2009.
<<https://elzevir.fr/imj/latex/cies.html>>.
Un cours très précis et détaillé sur des mécaniques importantes en \LaTeX .
- [31] Christian ROLLAND : *\LaTeX par la pratique*. O'Reilly & Associates, 1994.
Une bonne introduction qui, si elle est ancienne, a le mérite de présenter la plupart des fonctionnalités pratiques, y compris les aspects graphiques.
- [32] Maïeul ROUQUETTE : *Geekographie Maïeulesque*.
<<http://geekographie.maieul.net/LaTeX>>.
Ce blog présente de nombreux trucs et astuces pour aller plus loin sur l'utilisation des bibliographies et des index avec \LaTeX .
- [33] Thomas THURNHERR : *\TeX Blog*.
<<http://texblog.org/>>.
Un blog actif en anglais sur \LaTeX .
- [34] Mark TRETIN : *Une liste des péchés des utilisateurs de \LaTeX 2_ε*. 2004.
<<http://www.lsv.ens-cachan.fr/~markey/LaTeX/doc/l2tabufr.pdf>>.
Une fois que vous aurez roulé votre bosse sur le sujet de \LaTeX , pensez à consulter ce document pour vérifier que vous n'avez pas des réflexes un peu datés ou « exotiques ».
- [35] Marc R.C. van DONGEN : *\LaTeX and friends*. Springer, 2012.
 \LaTeX reste un sujet d'actualité, preuve en est cet ouvrage en anglais paru en 2012 faisant le point sur le sujet. Il détaille en particulier pour les graphiques le paquet `tikz`.