Introduction à T_EX

MANUEL D'AUTO-FORMATION

Michael Doob
Département de Mathématiques
Université de Manitoba
Winnipeg, Manitoba, Canada R3T 2N2

MDOOB@UOFMCC.BITNET

MDOOB@CCU.UMANITOBA.CA

- ATTIC - Traduction & Interprétariat, 17, rue Hôtel des Postes ,F - 06000 Nice
- Logiciels du Soleil, 1 rue Pasqualini, F-06800 Cagnes sur mer

Introduction

Commençons par les mauvaises nouvelles : T_EX est un gros programme complexe doté de fonctions extraordinaires permettant de générer des documents dactylographiés de qualité. Sa complexité même peut conduire à des choses inattendues de temps en temps. Les bonnes nouvelles : les textes simples sont très faciles à saisir avec T_EX. Il est ainsi possible de commencer avec du texte simple et d'aborder ensuite des tâches plus compliquées.

Le but de ce manuel est de commencer par le commencement et de progresser vers des tâches plus difficiles. Aucune connaissance préalable de T_EX n'est nécessaire. En procédant section par section, il est possible de produire une plus grande variété de texte.

Voici quelques suggestions : chaque section comprend des exercices. Faites-les ! La seule façon d'apprendre T_EX, c'est de l'utiliser. Mieux encore, en faisant vos propres expériences et en variant vous-même les exercices. Il vous sera impossible d'endommager le programme T_EX en faisant ces expériences. Vous obtiendrez une réponse complète à la plupart des exercices en regardant le fichier source T_EX utilisé pour produire ce document. Vous remarquerez qu'il existe des références dans la marge droite du **The T_EXBook**[1]. Vous devrez vous y référer lorsque vous rechercherez des informations.

De temps en temps, vous découvrirez quelques simplifications dans ce manuel; elles sont utilisées pour dissimuler les complications (je trouve que cela a quelque chose de poétique). Grâce à votre expérience de l'utilisation de T_EX, vous serez à même de les découvrir.

T_EX est un programme du domaine public disponible sans droit de licence. Il a été développé par Donald Knuth de l'Université de Stanford dans le cadre d'un important projet. Dans un marché basé sur le profit, ce programme coûterait certainement des milliers de dollars. Le Groupe d'Utilisateurs T_EX est une organisation à but non lucratif qui distribue des copies de T_EX, des versions actualisées et fournit des informations sur les nouveaux développements matériels et logiciels dans ses publications TUBboat et T_EX niques. L'adhésion à ce groupe est bon marché; pensez-y. L'adresse est la suivante :

T_EX Users Group
P.O. Box 9506
Providence, RI 02940
U.S.A.

[1] Addison Wesley, Reading, Massachusetts, 1984, ISBN 0-201-13488-9

TABLE DES MATIERES

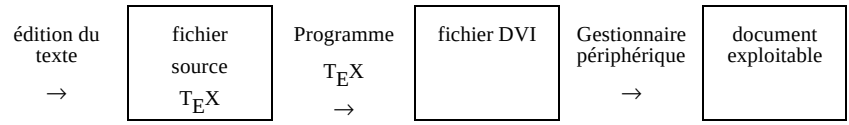
---

1 DEMARRAGE

1.1 Présentation de T_EX

Commençons d'abord par les premières étapes nécessaires à la production d'un document avec T_EX. La première étape est de saisir le fichier que T_EX lit. Il est généralement appelé fichier T_EX ou fichier d'entrée, et il peut être créé en utilisant une simple éditeur de texte (en fait, lorsque vous utilisez un traitement de texte évolué, vous devez vous assurer que le texte est bien sauvegardé en ascii ou en mode texte sans caractères de contrôle spéciaux). Le programme T_EX lit ensuite votre fichier d'entrée et produit ce que l'on appelle un fichier DVI (DVI signifie DeVice Independent). Ce fichier n'est pas lisible, du moins par les humains. Le fichier DVI est ensuite lu par un autre programme appelé gestionnaire de périphérique) qui produit la sortie qui est lisible par les humains. Pourquoi ce fichier supplémentaire ? Le même fichier DVI peut être lu par différents gestionnaires de périphériques et imprimé sur une imprimante matricielle, une imprimante laser, un éditeur d'écran* ou une photocomposeuse. Dès que vous aurez produit un fichier DVI qui sort correctement sur un éditeur d'écran par exemple, vous aurez la certitude que vous obtiendrez exactement la même chose sur une imprimante laser sans devoir utiliser à nouveau le programme T_EX.

Ce processus peut être illustré de la manière suivante :



Cela implique que vous ne verrez pas notre sortie dans sa forme finale quand il sera saisi sur le terminal. Cependant, votre patience sera largement récompensée étant donné qu'un grand nombre de symboles non disponibles sur les traitements de texte seront disponibles. En outre, la composition est réalisée avec une plus grande précision et les fichiers d'entrée peuvent être facilement transmis entre différents ordinateurs par courrier électronique ou sur un support magnétique.

Nous développerons la première étape, en l'occurrence la création du fichier d'entrée puis le lancement du programme T_EX permettant de produire les résultats appropriés. Il existe deux façons de lancer le programme T_EX; il peut être exécuté en mode batch ou de manière interactive. En mode batch, vous soumettez votre fichier d'entrée T_EX à votre ordinateur et il exécute le programme T_EX sans autre intervention et vous donne ensuite le résultat. En mode interactif, le programme peut s'arrêter et accepter une saisie supplémentaire de l'utilisateur, c'est à dire que l'utilisateur peut interagir avec le programme. L'utilisation interactive de T_EX permet à l'utilisateur de corriger certaines erreurs; tandis que le programme effectue les corrections en mode batch de la meilleure manière possible. Le mode interactif est naturellement le mode privilégié. Les applications de tous les ordinateurs personnels et de nombreux gros ordinateurs sont interactives. Sur certains gros ordinateurs, cependant, la seule méthode pratique d'exécution du programme T_EX est le mode batch.

1.2 Du fichier T_EX à la sortie exploitable : procédure

[Note de MD : ceci représente la seule section relative au système dans ce manuel et elle peut être remplacée par un guide spécifique. Aucune référence n'y est faite en dehors de la section elle-même. Les informations spécifiques suivantes devraient être ajoutées :

- Quelles étapes initiales éventuelles devraient être prises par le lecteur pour permettre le lancement de TEX et de votre gestionnaire(s) de périphérique.
- Comment exécuter TEX
- Comment lire le journal
- Comment visualiser et/ou imprimer le fichier DVI

L'exemple suivant est applicable ici, à l'Université de Manitoba. Nous utilisons un éditeur conçu sur place (MANTES) sur un Amdahl utilisant MVS. Je suis quasiment persuadé que c'est la pire des situations...

Dans ce chapitre, nous verrons comment exécuter TEX à l'Université de Manitoba. On suppose que le lecteur est déjà familiarisé avec MANTES et qu'il sait l'utiliser pour éditer des fichiers texte.

Tout d'abord, il y a certaines choses qui doivent être effectuées une par une. Pour démarrer, vous devez faire les choses suivantes (vous devez taper ce qui ressemble à des caractères de machine à écrire) :

(1) Allouez les fichiers que TEX utilisera en tapant les lignes suivantes (avec MANTES)

C:alloc da=source.tex format=vb,256,6144

C:alloc da=dvi format=fb,1024,6144

(2) Créer un fichier appelé RUNTEX dans votre agrégat MANTES* contenant le JCL suivant :

// JOB , 'RUN TEX'

// EXEC TEXC

// SRC DD DSN=<userid>.SOURCE.TEX, DISP=SHR

// DVI DD DSN=<userid>. DVI,DISP=OLD

Le nom <userid> est naturellement remplacé par votre propre identificateur. L'utilisation des majuscules et des espaces doit être rigoureusement respectée.

(3) Créez un fichier appelé PRINTTEX dans votre agrégat MANTES contenant le JCL* suivant :

// JOB ,'PRINT TEX'

// EXEC TEXP

// DVIFILE DD DSN=<userid>.DVI, DISP=SHR

Dès que ces trois étapes sont terminées, vous êtes prêt à lancer une application TEX. Les fichiers que vous avez créés vous permettront de produire environ 10 pages de texte ordinaire.

Voici les différentes étapes que vous devez suivre **à chaque fois pour lancer une application**

(1) créez un fichier MANTES contenant votre entrée TEX.

(2) sauvegardez et soumettez votre fichier en utilisant les commandes suivantes :

C:\save f/l to da=source.tex noseq

C:submit runtex

(3) dès que vous obtenez un message indiquant que votre tâche est terminée, entrez la commande suivante :

C: out <jobname>; list ttyout

Dans cette commande, <jobname> est remplacé par votre identificateur accompagné d'un signe dollar. C'est une version abrégée de ce que l'on appelle un "journal"; nous utiliserons le terme "journal" en référence au fichier ttyout produit par TEX.

Si vous le désirez, vous pouvez vérifier le statut de votre tâche en cours d'exécution en utilisant la commande :

C: q<jobname>

Lorsque vous avez fini de lire le journal, la commande end scratch supprimera le journal tandis que la commande end release enverra le journal à l'imprimante et il pourra ensuite être récupéré avec votre sortie TEX.

(4) lorsque votre sortie du programme TEX sera exempt d'erreurs, TEX créera un fichier DVI legal*. Pour l'imprimer, utilisez la commande :

C/ submit printtex nohold

Comme pour le (3), vous pouvez vérifier le statut de votre tâche en cours d'exécution.

(5) Récupérez votre sortie dans la fenêtre de saisie, sixth floor Engineering building. Il faut environ vingt minutes pour que la sortie soit prête. Appelez-le en utilisant "<jobname>".

Les fichiers créés sont suffisamment importants pour exécuter des tâches TEX d'environ 10 pages. Une tâche de cette importance prendra environ une seconde de temps de traitement du CPU. Il faudra environ 15 secondes de temps de traitement du CPU pour imprimer 10 pages sur une Xerox 8600 en utilisant le gestionnaire de périphérique actif.

Vous pouvez imprimer votre propre exemplaire du manuel en utilisant la commande docu tex. Vous pouvez également trouver une multitude d'informations en ligne sur TEX en utilisant %texinfo.

1.3 Application pratique

Ainsi, de notre point de vue, le but du jeu est de créer le fichier TEX produisant la sortie exploitable appropriée. A quoi ressemble un fichier TEX ? Il est constitué de caractères provenant d'un terminal ordinaire, c'est à dire de lettres majuscules et minuscules, de nombres, des signes de ponctuation et des caractères accentués (caractères ASCII habituels). Le texte, pour l'essentiel, est saisi normalement. Les instruction spéciales commencent habituellement par l'un des signes spéciaux tels que # ou & (ces derniers seront détaillés ultérieurement). Voici un exemple de fichier d'entrée TEX :

Here is my first \Tex\ sentence.

\bye

Notez tout d'abord que les caractères de cet exemple ressemblent à des caractères de machine à écrire. Nous utilisons ces caractères pour tous les exemples devant être saisis à partir du terminal de l'ordinateur. Ensuite, notez que le signe \ apparaît trois fois dans le texte. Nous verrons bientôt que ce dernier est l'un des symboles indiqués précédemment et qui est très souvent utilisé pour réaliser des documents T_EX. Constituez un fichier contenant cet exemple. Utilisez le programmes T_EX pour constituer un fichier DVI et un gestionnaire de périphérique pour visualiser ce que vous avez produit. Si tout se passe bien, vous obtiendrez une seule page avec la seule phrase suivante:

Here is my first T_EX sentence.

Il y aura un numéro de page au bas de la page. Si vous avez réussi à aller jusque là, bravo ! Dès que vous serez capable de produire un document T_EX, il vous suffira d'un peu de temps pour produire des document plus compliqués. Comparons maintenant ce que vous avez saisi et ce que vous avez obtenu. Les mots ont été simplement saisis normalement, et T_EX les inscrit en caractères ordinaires. Cependant, le mot "T_EX", qui ne peut être saisi sur un terminal étant donné que les lettres ne sont pas sur la même ligne, est saisi en utilisant un mot commençant avec un \, et T_EX l'a interprété correctement. La plupart des symboles qui ne sont pas des caractères, des nombres ou des signes de ponctuation ordinaires sont dactylographiés en écrivant un mot précédé d'un \. Si nous regardons de plus près, nous remarquons que le mot "first' est également modifié. Les deux premières lettres ont été jointes et il n'y a pas de point sur le "i". Ceci est une pratique courante d'imprimerie : certaines combinaisons de lettres sont jointes pour former ce qu'on appelle des *ligatures*. Il existe une bonne raison d'ordre esthétique à cela. Comparez les deux premières lettres de "first" et "first" pour voir la différence. Nous nous apercevons que \bye apparaît dans le fichiers d'entrée sans pour autant apparaître dans le document final. Ceci est une instruction d'impression qui indique à T_EX que la saisie est terminée. Nous apprendrons de nombreuses autres instructions en parcourant ce manuel.

Regardons tout d'abord le fichier journal créé lors de l'exécution de T_EX. Il se peut que sa forme varie en fonction de votre équipement mais il devrait ressembler à ce qui suit :

1.
2. This is Tex, MVS Version 2.9 (no format preloaded)
3. ** File PLAIN.FMT <-- DD=FMTLIB MEM=PLAIN
4.
5. ** File SRC.TEX <-- DD=SRC
6. (src.tex [1]
7. Output written on DVI (1 page, 256 bytes)
8. Tanscript written on TEXTLOG.

Ceci est le fichier qui contiendra les éventuels messages d'erreur. A la ligne 6, (scr.tex indique que T_EX a commencé la lecture du fichier). L'indication [1] indique que la page 1 a été traitée. S'il y avait des erreurs à la page 1, elles seraient mentionnées à ce niveau-là.

Ø Exercice 1.1 Ajoutez une deuxième phrase à votre fichier T_EX d'origine pour obtenir :

Here is my first \Tex\ sentence.

I was the one who typeset it !

\bye

Utilisez T_EX et examinez le document produit. La deuxième phrase est-elle sur une nouvelle ligne ?

Ø Exercice 1.2 Ajoutez cette ligne au début de votre fichier. :

\nopagenumbers

Devinez ce qui va produire lorsque vous exécuterez T_EX. Essayez maintenant et regardez ce qui se produit.

Ø Exercice 1.3 Ajoutez trois phrases ou plus à votre fichier. Utilisez des lettres, des nombres, des points, des virgules, des points d'interrogation et d'exclamation, mais n'utilisez pas d'autres symboles.

Ø Exercice 1.4 Laissez une ligne vierge et ajoutez des phrases. Vous pouvez maintenant obtenir de nouveaux paragraphes.

Nous venons de voir un principe fondamental de préparation des fichiers d'entrée T_EX. Le positionnement du texte sur le terminal de votre ordinateur ne correspond pas nécessairement au positionnement du texte de votre sortie. Vous ne pouvez pas, par exemple, ajouter des espaces entre les mots de votre sortie en ajoutant des espaces dans votre texte d'entrée. Plusieurs espaces consécutifs ou un seul espace produiront le même résultat. Comme prévu, un mot à la fin d'une ligne sera séparé du premier mot de la ligne suivante. En fait, lorsque l'on travaille sur un fichier enrichi, il est plus facile de commencer chaque phrase sur une ligne distincte. Les espaces en début d'une ligne, cependant, sont toujours ignorés.

Ø Exercice 1.5 Ajoutez la phrase suivante comme nouveau paragraphe et composez-la.

Congratulations! You received a grade of 100% on your latest examination.

Le signe % est utilisé pour les commentaires dans votre fichier d'entrée. Tout ce qui suit ce symbole sur une ligne est ignoré. Remarquez que même l'espace qui sépare normalement le dernier mot sur une ligne du premier mot est perdu. Insérez maintenant un \ en face du signe % pour corriger la phrase.

Ø Exercice 1.6 Ajoutez la phrase suivante comme nouveau paragraphe :

You owe me $10.00 and it's about time you sent it to me!

Ceci produira une erreur dans votre journal (si votre application de T_EX est interactive, c'est à dire qu'elle envoie des messages et attend des réponses, appuyez sur la touche retour de chariot ou Entrée lorsque vous obtenez le message d'erreur). Vous obtiendrez une sortie mais pas ce que espérez. Regardez le journal et repérez où sont indiqués les messages d'erreur. Ne vous souciez pas de ces messages. Nous développerons la question des erreurs (y compris celle-ci) ultérieurement). Corrigez maintenant l'erreur en utilisant un \ en face du signe $, et affichez le résultat (il y a un petit nombre de caractères tels que le signe % et $ que T_EX utilise à ses propres fins. Un tableau de ces caractères sera fourni plus loin).

1.4 T_EX contrôle tout

Nous venons de voir que \ a un rôle particulier. Tout mot commençant par \ sera traité à part lorsque T_EX le lira dans le fichier d'entrée. Un tel mot est appelé séquence de contrôle. Il existe en fait deux types de séquence de contrôle. Un *code de contrôle* est un \ suivi par des lettres exclusivement (par

exemple, \Tex) et un symbole de contrôle est un \ suivi d'un seul signe non alphabétique (par exemple, \$). Étant donné qu'un espace n'est pas un signe alphabétique, un \ suivi d'un espace est un symbole de contrôle légitime. Lorsque nous voudrons montrer qu'un espace est présent, nous utiliserons un symbole spécial u pour indiquer l'espace, tel que dans le symbole de contrôle \u. Cette convention est utilisée dans **The TEXbook** ainsi que dans ce manuel.

Lorsque TEX lit votre fichier d'entrée et arrive à un \ suivi d'une lettre, il sait qu'il s'agit d'un code de contrôle. Il continue donc de lire le nom du code de contrôle jusqu'à ce qu'un signe non alphabétique soit lu. Ainsi, si votre fichier contient

I like \TeX !

Le code de contrôle \TeX est terminé par un point d'exclamation. Mais ceci présente un problème si vous voulez avoir un espace après un code de contrôle. Si, par exemple, vous avez la phrase suivante

I like \TexT and use it all the time.

dans votre fichier d'entrée, le code de contrôle \TeX est terminé par l'espace (qui est, bien sûr, un signe non alphabétique). Mais alors, vous n'aurez plus d'espace entre les mots " TEX" et "and"; et l'insertion d'espaces supplémentaires ne changera rien, étant donné que TEX ne fait pas la différence entre un espace et une série d'espaces. Mais si vous mettez le signe de contrôle \u après un code de contrôle, vous terminez à la fois le code de contrôle et insérerez un espace. Il est très facile d'oublier d'insérer quelque chose comme \u après un code de contrôle. Je peux vous garantir que vous le ferez au moins une fois au cours de l'apprentissage de TEX.

Ø Exercice 1.7 Créez un fichier d'entrée qui produira la paragraphe suivant :

I like TEX ! Once you get the hang of it, TEX is really easy to use. You just have to master the TEXnical aspects.

la plupart des codes de contrôle sont nommés de telle sorte qu'ils donnent une idée de leur utilisation. Vous pouvez utiliser \par pour créer un nouveau paragraphe, par exemple, au lieu d'insérer une ligne vierge.

1.5 Ce que TEX ne peut pas faire

TEX est très efficace pour la composition, mais il y a des choses que TEX ne peut pas faire. L'une d'elle est la réalisation de dessins*. Des espaces peuvent être utilisés pour insérer des dessins* mais il n'existe aucune procédure graphique intégrée dans la langage (pour l'instant). Certaines applications permettent l'insertion d'instructions graphiques par l'utilisation du code de contrôle \special mais ces dernières sont exceptionnelles et spécifiques à chaque site.

TEX compose les caractères sur des lignes droites horizontales et non sur des lignes inclinées. Il est en particulier impossible de faire des insertions en "mode paysage", c'est à dire avec le texte pivoté de telle sorte que la ligne de base est parallèle au côté le plus long de la feuille, ni même d'inclure un texte dont la ligne de base est une courbe. Les caractères en perspective (augmentant ou diminuant progressivement de taille en hauteur) ne sont pas traités par TEX.

Nous avons vu qu'il existe un cycle "édition, TEX, périphérique" nécessaire à chaque exemplaire de sortie différent. Ceci est vrai même lorsque le périphérique de sortie est un terminal. En particulier, il n'est pas possible d'imprimer le texte d'entrée et de voir les résultats immédiatement à l'écran sans effectuer le cycle complet. Certaines applications sont dotées d'affichages texte et graphique relativement rapides (quelques secondes pour une seule page); étant donné que le matériel devient

moins cher et que les processeurs deviennent de plus en plus rapides, des améliorations sont prévisibles.

CHAPITRE 2

TOUS LES CARACTERES, PETITS ET GRANDS

2.1 Certains caractères sont plus spéciaux que d'autres.

Nous avons vu dans la dernière section que la plus grande partie du texte est saisi en tant que phrases composées de mots ordinaires comme si elles étaient dactylographiées avec une machine à écrire. Mais nous avons vu également que, en particulier, la barre descendante pouvait être utilisée pour au moins deux fonctions. Elle peut être utilisée pour des symboles (ou des combinaisons de symboles) qui n'apparaissent pas sur le clavier tels que \TeX pour TEX. Elle peut également être utilisée pour donner à TEX des instructions spéciales telles que \bue pour indiquer la fin du fichier d'entrée. En règle générale, un mot commençant par une barre descendante sera interprété par TEX comme une commande particulière. Il existe des centaines de mots connus de TEX, et vous pouvez en définir d'autres vous-même, et par conséquent ce signe est très important. Nous consacrerons beaucoup de temps à l'apprentissage de ce dernier; par chance, nous n'aurons à utiliser qu'un nombre limité de ces signes la plupart du temps.

Il existe dix caractères qui, à l'instar de la barre descendante, sont utilisés par TEX à des fins particulières, et nous voulons maintenant en donner une liste complète. Que se passe-t-il si nous voulons utiliser une phrase avec l'un de ces caractères spéciaux ? En ayant ceci à l'esprit, nous allons nous poser les questions suivantes :

(1) Quels sont les différents caractères spéciaux ?

(2) Comment allons-nous utiliser un caractère spécial si nous voulons vraiment l'insérer dans notre texte ?

Voici un tableau de chaque caractère spécial, sa fonction, et la méthode de saisie de ce caractère lorsque vous en avez besoin :

Caractères nécessitant une saisie particulière

Caractère	Fonction	Saisie pour sortie litérale
\	Symboles et instructions spéciales	$\barre descendante$
{	Groupe ouvert	$\{$
}	Groupe fermé	$\}$
%	Commentaires	\%
&	Tabulations et alignement de tableaux	\&
~	Espace insécable	\~{}
$	Début ou fin de texte mathématique	\$
^	Exposants mathématiques	\^{}
_	Indices mathématiques	_{}

#	Définition de symboles de remplacement	\#

2.2 Composition avec un accent

Vous allez maintenant commencer à exploiter les richesses de TEX ! Jusqu'à présent, nous avons simplement utiliser TEX pour rendre nos documents attrayants, mais nous allons commencer à réaliser des choses qui sont délicates ou irréalisales sur une machine à écrire. En particulier, nous allons regarder dès maintenant les accents. Comment faites-vous pour produire un accent quand il n'apparaît pas sur le clavier ? Tout comme le symbole TEX, il est nécessaire d'entrer un mot commençant par une barre descendante. Pour le mot "première", par exemple, il vous faut taper premi\`ere (il vous faudra peut-être rechercher le symbole accent grave* sur votre clavier mais il doit y figurer[2]). En général, pour mettre un accent sur une lettre, il est nécessaire de la faire précéder de la séquence de contrôle.

Voici quelques exemples :

Entrée TEX	Sortie TEX
\`a la mode	à la mode
r\`esum\`e	résumé
soup\cucon	soupçon
No\"el	Noël
na\"\iuve	naïve

Nous allons voir différents principes illustrés par ces exemples. La plupart des accents sont produits en utilisant un symbole de contrôle de forme semblable. Quelques uns d'entre eux sont produits par des codes de contrôle contenant une seule lettre. Leur utilisation nécessite une certaine précaution car un espace doit être utilisé pour terminer le code de contrôle. Si vous avez soup\ccon dans votre fichier, par exemple, TEX recherchera le code de contrôle ccon[3].

Remarquez aussi qu'il existe un code de contrôle \i. Il génère la lettre "i" sans point; ceci permet d'ajouter un accent sur la partie inférieure de la lettre. Il existe un code de contrôle analogue \j qui produit un "j" sans point pour l'utilisation d'accents.

Liste des accents devant être suivis immédiatement d'une lettre

Nom	Entrée TEX	Sortie TEX
grave	\`o	ò
aigu	\'o	ó

[2] Si vous avez un vieux clavier et que l'accent n'y figure pas, vous pouvez utiliser \lq{}. De la même manière, \rq{} peut être utilisé pour le symbole '. Vous pouvez mémoriser les symboles sous la forme "left quote" et "right quote". En outre, \lq\lq{} mais cette méthode ne fonctionnera pas pour produire un accent sur la lettre suivante. Il est donc préférable d'avoir un clavier adapté.

[3] Nous étudierons une autre méthode lorsque nous aborderons la notion de groupement dans le chapitre 4.

circonflexe	\^o	ô
trémat	\"o	ö
tilde	\~o	õ
macron	\=o	ö
point	\.o	ö

T_EX permet également la composition de certains caractères autres qu'anglais

3. Nous verrons qu'il existe une autre méthode quand nous étudierons le concept de groupage dans le chapitre 4

Liste des symboles de langues étrangères disponibles

Exemple	Entrée Tex	Sortie Tex
Ægean, æsthetics	\AE, \ae	Ææ
Œuvres Hors d'œuvre	\OE, \oe	Œ, œ
Ångstrom	\AA, \aa	Åä
Øre, København	\O, \o	Øø
Nuß	\ss	ß
¿Si?	?`	¿
¡Si!	!`	¡
Señor	\~	ñ
	{\it\$	£

2.3 Points, tirets, guillemets, ...

La saisie a toujours été un compromis : le petit nombre de touches (par rapport au nombre de caractères disponibles) a modifié le travail du dactylographe. La préparation d'un document avec TEX ne suppose par autant de restrictions. Dans ce chapitre, nous allons voir les différences entre la dactylographie et l'utilisation de TEX.

Il existe quatre types de tirets. Le tiret de liaison est utilisé pour combiner les mots en une seule unité comme dans belle-mère. Le tiret numérique* est utilisé pour indiquer une séquence de pages, de nombres, d'années, etc... Le tiret syntaxique est un symbole grammatical. Le signe moins est utilisé pour les nombres négatifs. Voici leur utilisation :

Différents types de tirets

Nom	Entrée TEX	Sortie TEX	Exemple
Tiret de liaison	-	-	the space is 3-dimensional
Tiret numérique	- -	--	See pages 3-4
Tiret syntaxique	- - -	---	I saw them - there were 3 men alive
Signe négatif	**$--$**	--	The temperature dropped to -3 degrees

Ø Exercice 2.12 I entered the room and-horrors-I saw oth my father-in-law and my brother-in-law.

Ø Exercice 2.13 The winter of 1484-1485 was one of discontent.

Une autre différence entre la dactylographie et l'utilisation de TEX est l'utilisation de guillemets. Ils guillemets d'introduction et de conclusion de citation sont identiques sur une machine à écrire. Ils sont produits par TEX en utilisant l'apostrophe montante ou descendante. Un guillemet d'introduction est produit par ` et le guillemet de conclusion par ’. Remarquez qu'il n'est pas nécessaire d'utiliser la marque de citation habituelle (elle produit des guillemets de citation, en principe).

Ø Exercice 2.14 His “thoughtfulness” was impressive.

Ø Exercice 2.15 Frank wondered, “Is this a girl that can't say `No`?”

Les ellipses (trois points) sont parfois utilisées pour indiquer l'écoulement du temps ou des informations manquantes. La saisie de trois points produira des points très serrés. Les points séparés par des espaces normaux seront générés en insérant \dots dans votre fichier d'entrée.

Ø Exercice 2.16 He thought, “...and this goes on forever, perhaps to the last recorded syllable.”

L'autre problème avec les points est que le point après une abréviation est suivi d'un espace plus petit que celui qui suit le point à la fin d'une phrase. Il existe toujours deux façons de remédier à cela : Le point peut-être suivi soit de \u soit de ~ pour changer l'espacement. La deuxième solution produira un espace insécable; quand un espace de ce type se trouve entre deux mots, ces mots doivent être composés sur la même ligne. La saisie de Prof.~Knuth fixera ces deux mots sur une même ligne. Ceci est souhaitable pour des noms tels que Vancouver, B. C. et Mr. Jones pour que "Mr." et Jones ne soit pas sur des lignes séparées. Remarquez que la barre descendante n'est pas utilisée pour l'espace insécable.

Ø Exercice 2.17 Have you seen Ms. Jones?

Ø Exercice 2.18 Pr. Smith and Dr. Gold flew from L.A. to Washington, D. C., via Fargo, N.D.

2.4 Polices de caractères

La différence la plus évidente entre du texte dactylographié et un document produit par TEX est sans aucun doute les différentes polices de caractères ou types de symboles utilisés. Lorsque TEX est lancé, il dispose de 16 polices de caractères disponibles. Certaines d'entre elles sont utilisées à des fins techniques, mais néanmoins il existe différents types disponibles comme indiqué dans le tableau ci-dessous. Une liste complète des seize polices se trouve dans l'Annexe F de **The TEXBook**. La plupart est utilisée automatiquement; un indice mathématique, par exemple, est affiché en plus petits caractères par TEX sans intervention particulière de l'utilisateur.

Pour passer des caractères (romans) habituels aux caractères italiques, on utilise le code de contrôle \it. Pour revenir aux caractères romans habituels, il faut utiliser \rm. Par exemple, vous pourriez avoir la phrase suivante : J'ai commencé avec des caractères romans, \it je suis passé aux caractères italiques, \rm et je suis revenu aux caractères romans. Ceci devrait produire la phrase suivante : J'ai commencé avec des caractères romans, *je suis passé aux caractères italiques*[4], et je suis revenu aux caractères romans.

Des caractères autres qu'italiques sont également disponibles. Les caractères indiqués ci-dessous sont les plus fréquemment utilisés. Ils sont immédiatement disponibles au démarrage de TEX.

Échantillons de polices de caractères

Nom de la police	Séquence de permutation TEX	Exemples
Roman	\rm	caractères romans
Gras	\bf	**caractères gras**
italique	\it	*caractères italiques*
Incliné	\sl	caractères inclinés
Courrier	\tt	caractères courrier
Symboles math[5].	\cal	caractères manuscrits

Les polices inclinées et italiques semblent similaires au premier abord. La différence entre la lettre a de chaque exemple est flagrante. Lors de la permutation d'une police inclinée ou italique vers une police romane, la dernière lettre de la première police s'inclinera vers la première lettre de la police romane; les lettres semblent alors un peu comprimées, et pour compenser cet effet, il est possible d'ajouter un espace supplémentaire appelé correction italique. Il suffit pour cela d'utiliser le symbole de contrôle\/. Dans la phrase suivante, il n'y a pas de correction italique après la première séquence et lettres italiques mais il y en a une après la deuxième séquence. Vous pouvez constater la différence. Si la correction italique n'est pas utilisée, les lettres sont tassées, mais si la correction est utilisée, l'espacement est meilleur. Il n'est pas nécessaire de faire une correction italique quand les caractères italiques sont suivis d'une virgule ou d'un point, mais votre texte sera plus présentable si vous l'utilisez avant des marques de citation ou des parenthèses.

---

[4] Remarquez que la virgule et ce bas de page sont curieusement en italique. Nous verrons une autre méthode de changement des polices quand nous étudierons les groupements dans le chapitre 4.

[5] Cet exemple est trompeur car il est nécessaire de connaître la saisie de formules mathématiques pour l'utiliser, mais je tenais à l'indiquer. Vous pourrez utiliser ces lettres après avoir étudié l'utilisation des formules mathématiques.

Il est possible d'utiliser des polices autres que celles définies initialement par TEX si vous le souhaitez (à la condition qu'elles soient disponibles sur votre ordinateur, bien sûr). Différentes tailles peuvent être utilisées en utilisant le code de contrôle \magstep. Pour définir votre nouvelle police, vous devrez connaître son nom sur votre ordinateur. Par exemple, la police romane est appelée "cmr10" sur la plupart des systèmes. Si vous utilisez \font\bigrm = cmr0 scaled \magstep 1, vous pouvez alors utiliser \bigrm de la même façon que vous utiliser \it ou \rm.

La permutation par \bigrm donnera des caractères romans 20% plus grands que les caractères habituels. _font\bigbigrm = cmr = 10 scaled \magstep 2 définira une police environ 44% plus grande que les caractères romans habituels. On peut faire varier les tailles de \magstep 0 à \magstep 5. Sur la plupart des ordinateurs, \magstephalf est également disponible. Ceci représente un agrandissement d'environ 9,5%. Voici quelques exemples des différentes tailles :

Différents grossissements des polices

\magstep 0	Texte avec magstep 0
\magstephalf	Texte avec magstephalf
\magstep 1	Texte avec magstep1
\magstep 2	Texte avec magstep 2
\magstep 3	Texte avec magstep 3
\magstep 4	Texte avec magstep 4
\magstep 5	Texte avec magstep 5

Il est aussi possible d'utiliser des types de caractères entièrement nouveaux. Ceci dépend des polices disponibles sur les systèmes informatiques. De nombreux systèmes possèdent un fichier appelé CMSS10 contenant des caractères Sans Sérif. L'utilisation de \font\sf = cmss10 permettra d'utiliser le code de contrôle \sf de la même manière que \bf. Ceci étant précisé, la saisie de

\sf Voici un échantillon de notre nouvelle police Sans Sérif

donnera

Voici un échantillon de notre nouvelle police Sans Sérif***

Ø Exercice 2.19 Quel problème peut se poser si nous utilisons \ss au lieu de \sf pour obtenir la police Sans Sérif ? Indice : si la réponse de vous vient pas immédiatement à l'esprit, pensez à l'alphabet allemand et vous trouverez.

Ø Exercice 2.20 Composez un paragraphe en texte agrandi Sans Serif.

Les polices supplémentaires peuvent varier d'un site à un autre. Le tableau suivant indique les polices les plus couramment utilisées.

Noms des polices de caractères externes

CMBSY10	CMBXSL10	CMBXTI10	CMBX10	CMBX12	CMBX5
CMBX6	CMBX7	CMBX8	CMBX9	CMBX10	CMCSC10
CMDUNH10	CMEX10	CMFF10	CMFIB8	CMFI10	CMITT10
CMMIB0	CMMI10	CMMI12	CMMI5	CMMI6	CMMI7
CMMI8	CMMI9	CMR10	CMR12	CMR17	CMR5
CMR6	CMR7	CMR8	CMR9	CMSLTT10	CMSL10
CMSL12	CMSL8	CMSL9	CMSSBX10	CMSSDC10	CMSSI10
CMSSI12	CMSSI17	CMSSI8	CMSSI9	CMSSQI8	CMSSQ8
CMSS10	CMSS12	CMSS17	CMSS8	CMSS9	CMSY10
CMSY5	CMSY6	CMSY7	CMSY8	CMSY9	CMTCSC10
CMTEX10	CTEX8	CMTEX9	CMTI10	CMTI12	CMTI7
CMTI8	CMTI9	CMTI10	CMTI12	CMTT8	CMTT9
CMU10	CMVTT10				

Voici un petit historique de ces noms. Les deux premières lettres CM signifient Computer Modern, le nom donné à cette famille de polices par son concepteur. Le nombre à la fin indique la taille des points : une police 10 points représente la taille standard, la taille 7 points est utilisée pour les indices, et la taille 5 est utilisée pour la taille normale pour les indices secondaires, la taille 12 points est 20% plus grande que celle de 10 points, etc. Si les lettres CM sont suivies de la lettre B (Bold), les caractères sont en gras. De même, la lettre R (Roman) indique une police romane, le I une police italique, CSC indique des petites majuscules, Sl des caractères inclinés, SS signifie Sans Sérif, SY symboles, et TT machine à écrire.

Ø Exercice 2.21 Recherchez les polices disponibles sur votre ordinateur et imprimer les lettres et les chiffres pour chacune d'elles.

Ø Exercice 2.22 La police CMR12 est 20% plus grande que la police CMR10. De même, \magstep 1 agrandit les caractères de 20%. Prenez un texte et imprimez-le en utilisant CMR12 puis CMR10 avec \magstep 1. Les résultats sont tout à fait différents !

CHAPITRE 3

MISE EN FORME

Dans ce chapitre, nous allons étudier la façon de produire des textes de présentations et de formes différentes. Il est possible d'utiliser simplement TEX avec les tailles par défaut, bien sûr, tout comme nous l'avons fait jusqu'à présent. Nos documents vont désormais être plus créatifs. En ce qui concerne la taille des différentes parties d'une page de texte, plusieurs unités de mesure peuvent être utilisées.

3.1 Unités

TEX peut mesurer la longueur en utilisant de nombreux types d'unité; les plus courantes sont le pouce, le centimètre, le point et le pica. Les abréviations correspondantes sont in, cm, pt, et pc. Le point est défini par l'équation 1 pouce = 72.27 points, et le pica par 1 pica = 12 points. Par conséquent, le point est relativement petit et correspond à la taille d'un point décimal. Voici une illustration des différentes tailles :

1 pouce : _____

1 centimètre : _____

20 points : _____

1 pica : ____

Les points sont donc utilisés pour effectuer des changements fins. Un pica correspond environ à la distance entre les lignes de base de lignes consécutives de texte normal (non agrandi). TEX est très précis quant aux dimensions. Sa plus petit unité interne est inférieure à 1/4 000 000 de pouce. C'est donc la résolution du périphérique de sortie qui déterminer la précision de la sortie.

Deux autres unités de tailles et de polices différentes sont parfois utiles. Le "ex" fait environ la hauteur d'un petit "x" et le "em" est légèrement plus petit que la largeur d'un "M" majuscule.

La forme de la sortie est généralement déterminée par des codes de contrôle. Ces derniers sont nombreux. Ils permettent un contrôle très fin du texte final. La plupart du temps, cependant, seul un petit nombre d'entre eux est nécessaire.

3.2 Mise en forme des pages

Le texte sur une page est constitué de trois parties élémentaires. L'en-tête au-dessus du texte principal comprend souvent un titre de chapitre, le titre d'une section, ou un numéro de page, et il peut changer de forme en fonction du type de page, paire ou impaire. En dessous de ce dernier se trouve le texte principal qui inclut les notes de bas de page. Enfin, il y a le bas de page qui peut comporter un numéro de page.

Dans les exemples que nous avons étudiés jusqu'à présent, l'en-tête est resté vierge. Le bas de page contenait soit un numéro centré, soit aucune indication si \nopagenumbers a été utilisé. Nous

développerons les en-têtes et les bas de page plus loin dans ce chapitre. Concentrons-nous, pour l'instant, sur le texte principal.

Pour finir une page et en commencer une autre, \vfill \eject peut être utilisé. Le code de contrôle \eject valide la fin d'une page, tandis que \vfill transfère tout espace vertical restant vers le bas de la page (vous pouvez essayer de supprimer \vfill pour voir comment fonctionne l'équivalent vertical des lignes horizontales justifiées).

La largeur horizontale active du texte sur la page est définie par le code de contrôle \hsize. Il peut être modifié, par exemple 4 pouces, par le code de contrôle \hsize = 4 in à tout moment en utilisant les méthodes décrites dans les chapitres suivants. La valeur de \hsize active lorsque le paragraphe est terminé détermine la largeur du paragraphe. Comme vous pouvez le constater dans ce paragraphe, la largeur du texte peut être changée (dans ce cas précis 4 pouces) pour un seul paragraphe. Étant donné qu'elle représente la valeur active de la largeur horizontale, des expressions telles que \hsize = 0.75\hsize peuvent prendre une nouvelle valeur qui est relative à l'ancienne valeur.

L'équivalent vertical de \hsize est \vsize, qui est la hauteur active du texte principal. Elle peut être réinitialisée, tout comme \hsize. Ainsi, \vsize = 8 in peut être utilisé pour changer la hauteur verticale du texte principal. Notez que \vsize est la taille du texte à l'exclusion de tout en-tête ou bas de page.

Le texte peut également être déplacé sur la page. Le coin supérieur gauche du texte est décalé de 1 pouce vers le bas et de 1 pouce vers la droite par rapport au coin supérieur gauche de la page. Les codes de contrôle \hoffset et \voffset sont utilisés pour déplacer le texte horizontalement et verticalement. Ainsi, \hoffset = .75 in et \voffset = -.5 in déplaceraient encore le texte de 0.75 pouces vers la droite et de 0.5 pouces vers le haut. La plupart du temps, il vous suffira de définir \hoffset, \voffset et \vsize au début de votre texte uniquement.

Codes de contrôle pour la taille de la page

Nom	Code de contrôle T_EX	Valeur par défaut de T_EX (en pouces)
Largeur horizontale	\hsize	6,5
largeur verticale	\vsize	8,9
décalage horizontal[6]	\hoffset	0
décalage vertical[5]	\voffset	0

* Le texte est initialement paginé à un pouce vers le bas et la droite du coin supérieur gauche de la page.

Ø Exercice 3.1 Dactylographiez un paragraphe de texte composé de plusieurs lignes. Prenez plusieurs exemplaires de votre paragraphe et insérer \hsize = 5 in avant l'un deux et \hsize = 10 cm en face du deuxième. Essayez d'autres valeurs pour \hsize.

Ø Exercice 3.2 Insérez \hoffset = .5 in et \voffset = 1 in avant le premier paragraphe de votre exercice précédent.

Ø Exercice 3.3 Reprenez l'exercice précédent et insérer \vsize = 2 in avant le premier paragraphe.

Dans le chapitre précédent, nous avons vu qu'il était possible d'utiliser des polices de taille supérieure en utilisant le code de contrôle \magstep. Il est également possible d'agrandir tout le document en une seule fois. Si \magnification = \magstep 1 apparaît en haut de votre document, alors tout le document sera agrandi de 20%. Les autres valeurs de \magstep peuvent également être utilisées. Il faut insister sur le fait que \magnification doit être impérativement utilisé avant même de taper le premier caractère. Cet agrandissement pose un problème d'unités; si le document doit être agrandi de 20% et que \hsize = 5 in apparaît dans le fichier d'entrée T_EX, la sortie du document définitif devra-t-elle comporter un \hsize de 5 pouces ou devra-t-il être agrandi de 20% à 6 pouces ? Sauf indication contraire, toutes les dimensions seront agrandies uniformément lorsque \magnification sera utilisé. Néanmoins, dans certains cas, il ne sera peut-être souhaitable que ceci se produise. Par exemple, vous voudrez peut-être laisser un espace précis de 3 pouces pour insérer une image. Dans ce cas, toute unité pourra être modifiée avec true de telle sorte que \hsize = 5 true fixera toujours la largeur de ligne à 5 pouces, indépendamment de l'agrandissement.

Ø Exercice 3.4 Insérer \magnification = \magstep 1 à la première ligne de vos fichiers d'entrée et regardez les modifications sur le document.

3.3 Mise en forme des paragraphes

Lorsque T_EX lit le texte devant être composé à partir du fichier d'entrée, il lit un paragraphe à la fois puis le compose. Cela signifie que de nombreux contrôles sont effectués pour la forme de chaque paragraphe et que de nombreuses précautions sont prises. Nous avons déjà vu que \hsize peut être

[6] Ce texte a à l'origine une margue supérieur et une marge de gauche de 1 pouce.

utilisé pour contrôler la largeur d'un seul paragraphe. Mais imaginez un instant que les informations suivantes apparaissent dans votre fichier d'entrée

\hsize = 5 in

four score and seven years ...

.

.

.

... from this earth

\hsize = 6.5 in

Quelle est la largeur du paragraphe ? Le \hsize a été modifié au début du paragraphe et de nouveau à la fin. Etant donné que la paragraphe n'a pas été terminé (par l'insertion d'une ligne vierge ou \par) avant la deuxième modification, le paragraphe sera composé avec une largeur de 6,5 pouces. Cependant, si une ligne vierge était insérée avant \hsize = 6.5 in, le paragraphe serait composé avec une largeur de 5 pouces. Ainsi, en général, quand un paragraphe est défini, les valeurs des paramètres actifs lorsque le paragraphe est terminé sont les seules utilisées.

Voici un tableau avec divers paramètres de paragraphe

Exemples de paramètres de mise en forme de paragraphe

Fonction	Code de contrôle de TEX	Valeur par défaut de TEX
largeur	\hsize	6,5 pouces
alinéa 1ère ligne	\parindent	20 points
interligne	\baselineskip	12 points
distance entre paragraphes	\parskip	0 point

Le code de contrôle \noindent peut être utilisé au début d'un paragraphe pour éviter l'alinéa automatique sur la première ligne. Le code de contrôle affectera uniquement le paragraphe défini lorsqu'il est concerné*. (naturellement, \parindent = 0 pt ne créera aucun alinéa sur les paragraphes).

Une manière plus souple de contrôler la largeur d'un paragraphe est d'utiliser \rightskip et \leftskip. La séquence \leftskip = 20 pt déplace la marge gauche du paragraphe de 20 points supplémentaires. Les valeurs négatives peuvent être assignées à \leftskip pour déplacer la marge gauche vers l'extérieur. Le code de contrôle rightskip produit la même chose pour le côté droit du paragraphe. Le code de contrôle \narrower produit la même chose que le paramétrage de \leftskip et de \rightskip sur la même valeur que \parindent. Ceci est très utile pour les longues citations, et ce paragraphe est un exemple de son utilisation. A l'instar de \hsize, la valeur active de \leftskip et de \rightskip, lorsque le paragraphe est terminé, est celle qui s'appliquera au paragraphe entier.

Ø Exercice 3.5 Faites deux paragraphes avec les spécifications suivantes : la marge gauche des deux paragraphes est décalée de 1,5 pouces, la marge droite du premier paragraphe est décalée de 0,75 pouces, et la marge droite du second paragraphe est décalée de 1,75 pouces.

Les lignes peuvent être composées de différentes longueurs à l'intérieur d'un même paragraphe en utilisant \hangindent et \hangafter. L'importance de l'alinéa est définie par la valeur de \hangindent. Si \hangindent est positif, l'alinéa est effectué à partir de la gauche, et s'il est négatif, il est effectué à partir de la droite. Les lignes sur lesquelles l'alinéa survient est contrôlé par \hangafter. Si \hangafter est positif, il définit le nombre de lignes sur toute la largeur avant le début de l'alinéa. Ainsi, si \hangindent = 1,75 in et \hangafter = 6, alors les six premières lignes occuperont toute la largeur et l reste sera décalé de 1,75 pouces à partir de la gauche. D'un autre côté, si \hangindent = -1,75 in et \hangafter = -6 alors les six premières lignes seront décalées de 1,75 pouces à partir de la droite et les reste occupera toute la largeur. TEX réinitialise les valeurs par défaut \hangindent = 0 pt t \hangafter = 1 après chaque paragraphe. Ces codes de contrôle sont utiles pour les paragraphes avec des "alinéas en sommaire" et pour entourer un espace libre réservé à une image. Le code de contrôle \hang au début du paragraphe produira une ligne sur toute la largeur (\hangafter = 1) et le reste du paragraphe sera décalé de la valeur active de \parindent. Cependant, vous devez utiliser \niindent si vous voulez que la première ligne s'étende jusqu'à la marge gauche.

Voici le paragraphe précèdent reproduit avec \hangafter et \hangindent = -1,75 in.

Les lignes peuvent être de différentes longueurs dans un même paragraphe en utilisant \hangindent et \hangafter. L'importance de l'alinéa sera déterminée par la valeur de \hangindent. Si \hangindent est positive, l'alinéa est effectué à partir de la gauche, et s'il est négatif, il est effectué à partir de la droite. La ligne sur laquelle l'alinéa intervient est contrôlée par \hangafter. Si \hangafter est positif, alors il déterminera le nombre de lignes occupant toute la largeur avant que n'intervienne l'alinéa.

Ainsi, si \hangindent = 1.75 in et \hangafter = 6, alors les six premières lignes occuperont toute la largeur et le reste sera justifié à 175 pouces à partir de la gauche. D'autre part, si \hangindent = -1.75 in et \hangafter = -6, alors les six premières lignes seront justifiées de 1,75 pouces à partir de la droite et le reste occupera toute la largeur. TEX réinitialise les valeurs par défaut \hangindent = 0 pt et \hangafter = 1 après chaque paragraphe. Ces codes de contrôles sont utilisés pour les paragraphes avec des "alinéas en sommaire" et pour entourer un espace réservé à une figure. Le code de contrôle **hang** au début d'un paragraphe fera en sorte que le paragraphe occupe toute la largeur de la ligne (\hangafter=1) et le reste du paragraphe sera justifié selon la valeur de \parindent. Mais il n'est pas nécessaire d'utiliser **noindent** si vous voulez que la première ligne occupe toute la largeur jusqu'à la marge gauche.

Voici le paragraphe précédent reproduit avec \hangafter = -6 et \hangindent = -1,75 in.

Les lignes peuvent être réalisées avec différentes longueurs dans un même paragraphe en utilisant \hangindent et \hangafter. L'importance de la justification est déterminée par la valeur de

\hangindent. Si \hangindent est positif, la justification est effectuée à partir de la gauche, et si elle est négative, elle est réalisée à partir de la droite. Les lignes sur lesquelles la justification est effectuée sont contrôlées par \hangafter. Si \hangafter est positif alors il détermine le nombre de lignes de largeur complète avant que ne commence la justification. Ainsi, si \hangindent = 1.75 in et \hangafter = 6, alors les six premières lignes occuperont toute la largeur et le reste sera justifié à 1,75 pouces à partir de la gauche. Par ailleurs, si \hangindent = -1.75 in et \hangafter = -6 alors les six premières lignes seront justifiées de 1,75 pouces à partir de la droite et le reste occupera toute la largeur. TEX réinitialise les valeurs par défaut \hangindent = 0 pt et \hangafter = 1 après chaque paragraphe. Ces codes de contrôle sont utiles pour les paragraphes avec des justifications « sommaires » et pour répartir un paragraphe autour d'un espace réservé à une image. Le code de contrôle \hang au début du paragraphe permet à la première ligne d'occuper toute la largeur (\hangafter=1) et le reste du paragraphe sera justifié en fonction de la valeur active de \parindent. Cependant, vous devez utiliser \noindent si vous voulez que la première ligne s'étende sur toute la largeur jusqu'à la marge gauche.

le code de contrôle \parshape peut être utilisé pour former un paragraphe avec une plus grande variété d'aspects.

Un autre code de contrôle utile pour définir les paragraphes est \item. Il peut être utilisé pour créer différents type de listes détaillées et il est activé en utilisant la syntaxe \item{...}. Ceci permet de créer des paragraphes avec chaque ligne justifiée par \parindent et, en outre, la première ligne sera repérée sur la gauche avec les indications figurant entre les accolades. Il est habituellement utilisé avec \parskip = 0 pt, étant donné que ce code de contrôle détermine l'espace vertical entre les différents éléments. Le code de contrôle \itemitem est le même que \item sauf que la justification est deux fois plus grande, c'est à dire deux fois la valeur de \parindent. Voici un exemple :

```
\parskip = 0pt \parindent = 30 pt

\noindent

Answer all the following questions:

\item{(1)} What is question1?

\item{(2)} What is question2?

\item{(3)} What is question3?

\itemitam{(3a)} What is question3a?

\itemitem{(3b)} What is question3b?
```

Produira :

Answer all the following questions:

(1) What is question 1?

(2) What is question 2?

(3) What is question 3?

(3a) What is question 3a?

(3b) What is question 3b?

Ø Exercice 3.6 Créez un paragraphe de plusieurs lignes et utilisez-le avec le code \item pour visualiser la « justification sommaire ». Prenez ce même paragraphe et utilisez-le avec différentes valeurs de \parindent et \hsize.

Regardons maintenant la manière d'insérer des espaces entre des paragraphes. Le code de contrôle \parskip est utilisé pour déterminer l'espace normalement inséré entre les paragraphes. Ainsi, si vous indiquez \parskip = 12 pt au début de votre fichier source T_EX, il y aura 12 points entre les paragraphes sauf instructions contraires. Le code de contrôle \vskip peut être utilisé pour insérer un espace vertical supplémentaire entre les paragraphes. Si \vskip 1 in ou \vksip 20 pt apparaît entre deux paragraphes, alors l'espace supplémentaire est inséré.

Certaines particularités de \vskip peuvent sembler étranges au premier abord. Si vous avez \vskip 3 in et que le saut commence à deux pouces à partir du bas de la page, le reste de la page est sauté, mais le pouce supplémentaire n'est pas sauté au sommet de la page suivante. En d'autres termes, \vskip **n'insérera pas d'espace sur les limites de page** En fait, \vskip 1 in n'aura absolument aucun effet s'il apparaît au sommet d'une page! Pour de nombreux types d'espacement vertical, ceci est tout à fait approprié. L'espace précédent un en-tête, par exemple, ne devra pas se poursuivre sur des limites de page.

Un phénomène semblable se produit au début de votre document. Si, par exemple, vous voulez une page titre avec le titre vers le milieu de la page, vous ne pouvez pas insérer l'espace en haut de la page en utilisant \vksip.

Mais que faut-il faire pour obtenir un espace en haut d'une page ? Vous pourriez commencer la page avec \u mais ceci a en réalité pour effet d'insérer un paragraphe d'une ligne contenant un blanc. Par conséquent, alors qu'aucun élément n'est réellement saisi, l'espace supplémentaire dû aux valeurs de \baselineskip et \parskip ajoutera un espace supplémentaire. Une méthode plus simple consiste à utiliser \vglue à la place de \vskip pour obtenir le résultat désiré. Ainsi, \vglue 1 in laissera un espace d'un pouce en haut de la page.

Nous pouvons noter en passant qu'il existe une autre méthode générale pour insérer des éléments en haut d'une page en utilisant les codes de contrôle \topinsert et \endinsert. Si \topinsert ... ou \endinsert est utilisé dans une page, les éléments entre \topinsert et \endinsert apparaîtront en haut de la page, si possible. Par exemple :

```
\topinsert

\vskip 1 in

\centerline{Figure 1}

\endinsert
```

est utile pour insérer des images dans une zone déterminée.

Il existe également certains codes de contrôle spéciaux pour effectuer des petits sauts verticaux : \smallskip, \medskip, et \bigskip. Voici la taille de chacun d'eux :

\smallskip : _____ \medskip : _____ \bigskip : _____

3.4 Mise en forme des lignes

Pour la plupart des documents, T_EX est efficace pour transformer un paragraphe en ligne. Cependant, il est nécessaire d'ajouter des instructions supplémentaires. Il est possible de forcer une nouvelle ligne en insérant \hfill \break dans votre fichier d'entrée. Il est également possible d'ajouter du texte sur une ligne en utilisant le code contrôle \line{...}; alors les éléments entre les accolades seront limités à une ligne (bien que le texte soit réparti sur toute la ligne et puisse être inesthétique). Le code de contrôle \leftline{...}, \rightline{...}, et \centerline{...} composera, respectivement, les éléments entre accolades sur une seule ligne sur la marge gauche, sur la marge droite, ou centré.

Ainsi,

\leftline {I'm over on the left.}

\centerline{I'm in the center.}

\rightline{I'm on the right}

\line{I just seem to be spread out all over the place.}

Produit les quatre lignes suivantes :

I'm on the left.

I'm in the center.

I'm on the right.

I just seem to be spread out all over the place.

D'autres types d'espacement peuvent être obtenus en utilisant le code de contrôle \hfil. Tout l'espace supplémentaire sur la ligne devra être accumulé à la position d'insertion de \hfil. Ainsi, si nous modifions notre exemple de la manière suivante : \line{I just seem to be spread out \hfill all over the place.}, nous obtiendrons :

I just seem to be spread out all over the place.

Si nous avons plus d'un \hfill, ils diviseront équitablement tout espace supplémentaire restant. Par conséquent, \line{left text \hfill center text\hfill right text.} produira :

left text center text right text

Ø Exercice 3.7 Composez la ligne suivante :

Bord gauche taquet gauche garde gauche centre garde droite taquet droit bord droit

Ø Exercice 3.8 Composez la ligne suivante avec deux fois plus d'espace entre « gauche » et « centre-droit » qu'entre « centre-droit » et « droit » :

gauche centre droit droit

Il est possible de déplacer horizontalement en utilisant \hskip de la même manière qu'avec \vskip.

Ø Exercice 3.9 Que se passe-t-il pour la ligne suivante ?

\line{\hskip 1 in ONE \hfill TWO \hfill THREE}

La justification droite peut être supprimée en utilisant le code de contrôle \raggedright.

3.5 Bas de page

La syntaxe générale pour créer des bas de page en utilisant TEX est la suivante : \footnote{...}{...}.

La marque de bas de page s'insère entre les premières parenthèses. Les marques disponibles sont \dag (†), \ddag (‡), \S (§), et \P (¶). Le texte du bas de page s'insère entre les deuxièmes parenthèses. L'utilisation du nombre de marques est légèrement moins évident. La note de bas de page[7] a été réalisée en utilisant \footnote{$\{\}^{21}$}{Voici la note de bas de page.} après le mot « footnote » dans le texte. Cette syntaxe est relativement complexe; nous verrons pourquoi il en est ainsi lorsque nous en saurons davantage sur les formules mathématiques. Pour le moment, nous pouvons le considérer simplement comme une manière d'effectuer le travail. Il est possible d'utiliser \rm dans le bas de page pour s'assurer que les bonnes polices apparaissent. Normalement, vous n'aurez pas besoin d'un espace entre le texte et le code de contrôle footnote.

Ø Exercice 3.10 Créez une page avec un bas de page de plusieurs lignes.

Ø Exercice 3.11 Créez une page avec deux bas de page différents sur la même page.

3.6 En-têtes et Lignes de pied

Les lignes pour le titre et les numéros de page qui s'inscrivent au-dessus et en dessous du texte principal sont produits en utilisant \headline={...} et \footline={...}.

Le principe est le même que celui de l'utilisation du code de contrôle \line{...} dans le texte habituel sur la page. \pageno est un code de contrôle utile qui représente le nombre de la page active. Ainsi, \headline={\hfil \tenrm Page \the\pageno} inscrira le numéro de la page dans le coin supérieur droit précédé par le mot « Page » (regardez maintenant l'angle supérieur droit de cette page). Il est plus sûr de nommer explicitement la police que vous voulez utiliser (dans ce cas \tenrm pour utiliser la police roman 10 points), étant donné que vous n'aurez jamais la certitude qu'une police particulière sera utilisée lorsque l'en-tête ou la ligne de pied est créé. Le code de contrôle \the prend la valeur interne du prochain mot si c'est un code de contrôle approprié, et l'imprime sous forme de texte. Vous pouvez également utiliser le code de contrôle \folio à la place de \the \pageno. La différence est que \folio vous donnera des chiffres romains lorsque \PAGENO sera négatif.

[7] Voici la note de bas de page

Vous pouvez aussi assigner des valeurs à \pageno si vous voulez que votre document utilise une séquence de page différente de la séquence habituelle. Les chiffres romains seront produits en utilisant des nombres négatifs; \pageno=-1 au début du document produira des chiffres romains dans les numéros de page.

Des en-têtes différents peuvent être produits pour des pages paires ou impaires de la manière suivante :

\headline={\ifodd \pageno {...}\else {...}\fil}

lorsque que le contenu entre le premier ensemble d'accolades est destiné aux pages de droite et le contenu entre le deuxième groupe d'accolades est destiné aux pages de gauche.

Ø Exercice 3.12 Changez les lignes de pied de telle sorte que le numéro de page soit centré avec un tiret de chaque côté.

3.7 Overfull et underfull boxes

L'une des expériences les plus frustrantes pour le nouvel utilisateur de TEX est l'occurrence de cadres . Elles sont dûment indiquées dans le journal et sont également affichées sur l'écran lorsque TEX est utilisé interactivement. Les overfull boxes sont aussi indiquées sur la sortie par un lingot (un gros rectangle noir semblable à ceci : G) dans la marge droite. D'autres termes abscons sont utilisés pour désigner ce lingot. Il apparaît même lorsqu'il n'y a rien d'anormal avec le fichier d'entrée TEX. Alors, pourquoi ce lingot apparaît-il et que peut-on en faire ?

Une bonne façon de visualiser la façon dont TEX organise une page est de considérer le document imprimé comme un ensemble de boîtes. Il y a deux types de boîtes : Hboxes et vboxes. La plupart du temps, elles correspondent à l'organisation du texte horizontal en lignes et les paragraphes verticaux en pages. En particulier, c'est l'espacement des mots dans une hbox correspondant à une ligne de texte qui provoque l'apparition du lingot.

Souvenez-vous que TEX lit le paragraphe entier avant de décider de la manière de le répartir en lignes. Ceci est plus efficace que de travailler sur une ligne à la fois, étant donné que le moindre changement sur une ligne risquerait de provoquer des changements catastrophiques plus loin dans le paragraphe. Lorsque les mots sont regroupés pour former une ligne, un espace est ajouté entre les mots pour justifier la marge droite. Les espaces importants entre les mots sont inopportuns. Une boîte hbox insuffisamment remplie implique qu'il y a trop d'espace entre les mots. Plus précisément, l'aspect d'une ligne est indiqué par un nombre entre 0 (parfait) et 1000 (horrible). Il existe un paramètre appelé \hbadness dont la valeur par défaut est 1000. Toute ligne dont l'aspect est supérieur à \hbadness est répertoriée comme hbox insuffisante. Si la valeur de \hbadness est augmentée, alors un plus grand nombre de hboxes sous-remplies sera dénombrée. En fait, \hbadness = 10000 supprimera complètement le dénombrement des hboxes sous-remplies. De même, si les mots doivent être serrés sur une ligne pour que les espaces soient très petits ou qu'ils débordent même dans la marge droite, il en résulte alors une hbox sur-remplie.

De la même manière, TEX permet parfois le dépassement d'une ligne au-delà de \hsize afin de produire une présentation plus équilibrée. Le code de contrôle \tolerance détermine cette occurrence. Si l'aspect d'une ligne est plus grand que \tolerance, TEX allongera la ligne en ajoutant un nouveau mot à l'extrémité de la ligne, même si elle est susceptible d'excéder la valeur de \hsize. Une ligne légèrement plus longue est créée sans qu'elle soit répertoriée. Le code de contrôle \hfuzz détermine la limite du dépassement. La valeur par défaut est \hfuzz = 0.1 pt. Une ligne sensiblement plus longue que \hsize pose évidemment un problème sérieux. TEX insère un lingot dans la marge pour

marquer ce point. Il est possible d'éviter les hboxes surchargées en augmentant la valeur de \tolerance. Avec \tolerance = 10000, il n'y aura jamais de boîtes surchargées ni de lingots. La valeur par défaut est \tolerance = 200.

La largeur du lingot est déterminé par le code de contrôle \overfullrule. L'insertion de \overfullrule = 0 pt dans votre fichier effacera toute impression des lingots. Les boîtes overfull seront toujours présentes, bien sûr, et elles seront probablement plus difficiles à repérer.

Nous voyons donc pourquoi les boîtes overfull et underfull sont indiquées; nous pouvons également changer l'indication en changeant les valeurs de \badness, \hfuzz, et \tolerance. En outre, une petite valeur de \hsize rendent plus difficile la composition des lignes et provoque un plus grand nombre d'indications de boîtes overfull et underfull. Ce sont des avertissements que vous pouvez ignorer à vos risques et périls !

Le fait d'ajouter de nouvelles possibilités de césure éliminera parfois une boîte overfull. TEX intègre la césure automatique et trouve habituellement les endroits appropriés pour effectuer la césure d'un mot. Cependant, il est possible d'ajouter des césures pour insérer des césures à de nouveaux endroits. Par exemple, la césure automatique n'insérera jamais de tiret dans le mot « database ». Si vous tapez date\-base, il permettra l'insertion d'un tiret après la seconde lettre « a » du mot. Plus généralement, si vous écrivez \hyphenation{data-base} au début de votre fichier d'entrée, alors toutes les occurrences du mot « database » accepteront la césure après la lettre « a ». Le journal indiquera les césures possibles sur la ligne contenant la boîte overfull ou underfull. Parfois, la meilleure solution pour une hbox overfull ou underfull résidera simplement dans l'édition judicieuse du document original.

Nos discussions ont porté sur la composition de caractères sur des lignes, c'est à dire la structure de la page horizontale. Il existe plusieurs équivalents verticaux. Les hboxes overfull et underfull indiquent l'aspect des mots sur les lignes. De même, les vboxes overfull et underfull sont indiquées lorsque les paragraphes sont réunis pour former des pages. Un grand tableau qui ne peut être coupé au milieu, par exemple, peut produire une vbox underfull qui est indiquée dans le journal lorsque la page en cours de composition est terminée. Le code de contrôle \vbadness fonctionne pour le placement vertical du texte de la même manière que \hbadness pour le texte horizontal.

Ø Exercice 3.13 Prenez quelques paragraphes et imprimez-les en utilisant différentes (petites) valeurs de \hsize pour voir quel type de boîtes overfull en résulte. Répétez cet exercice avec différentes valeurs de \hbadness, \hfuzz et \tolerance.

CHAPITRE 4

{GROUPES, {GROUPES, {ET ENCORE DES GROUPES}}}

Le concept d'assemblage de groupes permet à TEX de simplifier énormément les fichiers d'entrée. Un nouveau groupe commence avec le caractère { et se termine avec le caractère }. Les changements effectués dans un groupe prendront effet à la fin du groupe. Par exemple, si {\bf three boldface words} apparaît dans votre texte, l'accolade d'ouverture commence le groupe, le code de contrôle \bf produit des caractères gras, et l'accolade de fermeture termine le groupe. Lorsque le groupe se termine, la police active est celle qui était utilisée avant le commencement d'un groupe. Ceci est une manière (plus facile) d'obtenir quelques mots avec des polices différentes. Il est également possible d'obtenir des groupes insérés dans des groupes.

Autre exemple : les changements de taille peuvent être effectués dans des textes qui sont seulement temporaires. Par exemple :

```
{
\hsize = 4 in
\parindent = 0 pt
\leftskip = 1 in
produira un paragraphe qui a
.
.
.
(ceci une erreur facile à faire)
\par
}
```

> produira un paragraphe qui a quatre pouces de large avec un texte inséré d'un pouce dans le paragraphe indépendamment des paramètres actifs avant que ne commence le groupe. Après la fin de ce paragraphe, les paramètres précédents sont de nouveau actifs. Il faut noter qu'il est nécessaire d'inclure \par ou d'utiliser une ligne vierge avant l'accolade de fermeture pour terminer ce paragraphe car, dans le cas contraire, le groupe se terminerait et TEX reprendrait les anciens paramètres avant que le paragraphe ne soit effectivement composé (ceci est une erreur facile à faire).

Lorsqu'un code de contrôle (tel que \centerline) agit sur le texte qui le suit entre accolades, ce texte est implicitement dans un groupe. Ainsi, \centerline{\bf A bold type} produira une ligne centrée en

caractères gras, et le texte suivant cette ligne sera composée de caractères dont la police est celle qui était active avant que ne soit activée \centerline.

Le groupe vide {} est utile. Il permet par exemple de taper des accents sans lettre. Par exemple, \~{} imprime un tilde sans lettre en dessous. Il peut également être utilisé pour empêcher TEX d'absorber les espaces consécutifs. Ainsi, **j'utilise \Tex{} tout le temps** laissera un espace après « Tex » sur la sortie. C'est une variante de \u comme nous l'avons vu dans la section 1.

Les groupements peuvent également être utilisés pour éviter les espaces au milieu d'un mot comportant un accent. Par exemple, soup\cucon ou soup\c{c}on produiront le mot soupçon.

Ø Exercice 4.1 Changez les dimensions d'un paragraphe sur une page en utilisant le concept de groupement.

Ø Exercice 4.2 Les mathématiciens utilisent parfois le mot « iff » comme abréviation de « if and only if». Dans ce cas, il est préférable que le premier et le deuxième « f » ne soient pas joints comme une ligature. Comment faites-vous cela (il y a plusieurs solutions) ?

Il est très facile d'oublier d'associer correctement les accolades. L'effet peut être spectaculaire; si vous obtenez une sortie qui génère brutalement des caractères italiques dans tout le reste du document, des accolades non équilibrées en sont probablement la cause. Si vous avez une { supplémentaire, TEX vous indiquera un message dans le journal : (\end occurred inside a group at level 1). Une } supplémentaire produira le message ! Too many }'s.

Voici une astuce pour vous aider à contrôler les accolades dans des groupes plus complexes : mettez l'accolade d'ouverture sur une ligne à part et faites la même chose pour l'accolade de fermeture. Sil y a des accolades incluses dans les premières accolades, mettez-les également sur des lignes distinctes mais décalez-les de quelques espaces. Le texte à l'intérieur des accolades incluses peut également être décalé étant donné que TEX ignore tous les espaces au début d'une ligne. Les couples d'accolades apparaîtront clairement dans votre fichier source TEX. En fait, si votre éditeur est suffisamment puissant, vous pouvez créer d'abord les deux lignes avec les accolades et ensuite insérer le texte approprié à l'intérieur avec une justification automatique.

Ø Exercice 4.3 Dans le chapitre 2 nous avons changé les polices de la manière suivante : **I started with romantype, \it switched to italic type, \rm and returned to roman type** Produisez le même résultat en utilisant le concept de groupements.

CHAPITRE 5

NULLE ANGOISSE DES MATHS !

TEX excelle dans la composition de textes mathématiques. Les conventions sont nombreuses et complexes, et la capacité de TEX à les prendre en compte rend possible la production de textes mathématiques attrayants et de haute qualité. Si vous envisagez de produire des documents avec des symboles mathématiques, ce chapitre vous donnera toutes les bases nécessaires pour créer de beaux documents dans presque toutes les circonstances. TEX peut être utilisé sans connaissances mathématiques bien sûr, et si c'est votre objectif, les deux sous-chapitres suivants seront probablement suffisants pour vos besoins.

5.1 De nouveaux symboles à foison

L'insertion de texte mathématique est effectuée sous forme de texte ordinaire de deux manières possibles; en ligne, c'est à dire, comme des lignes de texte ordinaire, ou affiché, c'est à dire centré dans un espace entre le texte habituel. Le résultat au niveau de l'espacement et du positionnement des symboles peut être tout fait différent dans chaque cas. L'équation en ligne est visiblement différente de la même équation affichée :

$$\sum_{k=1}^{\infty} \frac{1}{k^2} = \frac{^2}{6}$$

Etant donné que l'espacement et les polices utilisés pour les formules mathématiques sont tout à fait différents de ceux d'un texte ordinaire, TEX doit savoir lorsqu'il s'agit de texte mathématique plutôt que de texte ordinaire. Cette fonction est assurée par $. Plus précisément, le texte mathématique est composé en ligne en entourant les symboles à insérer par des signes dollar simples : $...$, et il est composé affiché en entourant les symboles à insérer par des signes dollar doubles : $$...$$. Ainsi $x = y+1$ donnera : x = y +1 en ligne tandis que $$x = y+1.$$ affichera

$$x = y+1$$

L'espacement des textes mathématiques en ligne ou affichés est contrôlé entièrement par TEX. L'ajout d'espaces dans votre fichier d'entrée n'a absolument aucun effet. Et si vous avez besoin d'un espace ou de quelque texte au milieu d'un texte mathématique ? Vous pouvez ajouter du texte en l'insérant dans une hbox (ne vous souciez par de la définition d'une hbox pour l'instant : \hbox{...}. Ceci est particulièrement utile pour les textes mathématiques affichés. Ainsi, "x = y+1 avec y = x-1" peut être composé en utilisant $x=y+1 \hbox{ avec }y=x-1$. Notez les espaces de chaque côté du mot dans les accolades. Habituellement, il n'est pas nécessaire d'insérer des espaces dans les textes mathématiques, mais voici la liste des séquences de contrôle qui le permettent.

Ajout d'espace dans des textes mathématiques

Nom	Séquence de contrôle.	⌋←Taille→⌊
Double quadrat	\qquad	⌋ ⌊
Simple quadrat	\quad	⌋ ⌊
Espace	\u	⌋ ⌊
Espace épais	\;	⌋ ⌊
Espace moyen	\>	⌋⌊

Espace réduit	\,	⌋⌊
Espace réduit négatif	\!	⫿

Si vous examinez l'espace réduit négatif, vous remarquerez que, contrairement aux autres entrées, les deux bras se chevauchent. Ceci est dû au fait que l'espace négatif est dans la direction opposé, c'est à dire que tandis que les autres séquences de contrôle augmente la quantité d'espace entre deux symboles composés, l'espace réduit négatif diminue l'espace entre eux même s'il provoque le chevauchement.

Ø Exercice 5.1 Tapez le texte suivant : C(n,r) = n!/(n-r)!). Notez que l'espacement dans le dénominateur

Vous ne devez pas avoir de lignes vierges entre les signes dollar délimitant le texte mathématique. TEX suppose que tout le texte mathématique en cours de saisie est dans un seul paragraphe, et qu'une ligne vierge commence un nouveau paragraphe; par conséquent, ceci provoquera un message d'erreur. Ceci s'avère utile car l'une des erreurs les plus faciles à faire est d'oublier de mettre le(s) signe(s) dollar de fermeture après le texte mathématique. (je suis prêt à parier que vous le ferez au moins une fois pendant votre apprentissage de TEX); si TEX a permis l'insertion de plusieurs paragraphes entre les signes dollar, alors un signe dollar manquant risque de provoquer la prise en compte du reste du document comme texte mathématique.

La plus grande partie du texte mathématique est saisi exactement de la même manière pour la saisie en ligne que pour la saisie affichée (à l'exception des signes dollars, bien sûr). Les exceptions, à savoir l'alignement d'affichages multilignes et le positionnement d'équations dans la marge gauche ou droite seront étudiés dans la dernière partie de ce chapitre.

De nombreux signes mathématiques nouveaux peuvent apparaître dans la saisie des textes mathématiques. La plupart de ceux qui apparaissent sur le clavier peuvent être utilisés directement. Les symboles + - / ' | * < > (et) sont tous saisis directement. Les voici tels qu'ils apparaissent sous forme de texte mathématique : + - / ' | * < > ( ).

Ø Exercice5.2 Saisissez a+b=c-d = xy=w/z sous forme de texte mathématique en ligne et affiché.

Ø Exercice5.3 Saisissez l'équation (fg)'=f'g+fg' sous forme de texte mathématique en ligne et affiché.

De nombreux autres symboles, comme vous pouvez l'imaginer, sont des codes de contrôle prédéfinis. Toutes les lettres grecques sont disponibles.

Ø Exercice 5.4 Saisissez $\alpha+\beta=\gamma+\delta$ sous forme de texte mathématique en ligne et affiché

Ø Exercice 5.5 Saisissez $\Gamma(n)-(n-1)!$ sous forme de texte mathématique en ligne et affiché

Des accents sont parfois ajoutés au dessus ou au dessous des symboles. Les codes de contrôle utilisés pour les accents en mathématique sont différents de ceux utilisés pour du texte ordinaire. Les code de contrôle pour du texte ordinaire ne peuvent pas être utilisés pour du texte mathématique et vice-versa.

Accents mathématiques

$\hat{o}$	\hat o	$\breve{o}$	\check o	$\tilde{o}$	\tilde o
o'	\acute o	$\grave{o}$	\grave o	$\dot{o}$	\dot o
$\ddot{o}$	\ddot o	$\breve{o}$	\breve o	$\bar{o}$	\bar o
$\vec{o}$	\vec o	abc	\widehat {abc}	abc	\widetilde {abc}

Les opérateurs binaires combinent deux objets mathématiques pour obtenir au autre objet. L'addition et la multiplication, par exemple, combinent deux nombres pour en obtenir un autre, et ils représentent donc des opérateurs binaires. Lorsqu'un opérateur binaire tel que + ou x est saisi, un petit espace supplémentaire est ajouté de chaque côté. Voici une liste de certains des opérateurs binaires disponibles :

◊	\diamond
★	\star
⋅	\dot
×	\times
∗	\ast
∘	\circ
•	\bullet
÷	\div
∘	\diamond
∩	\cap
∪	\cup
∨	\vee
∧	\wedge
⊕	\oplus
⊗	\otimes

Les ellipses sont couramment utilisées avec les opérateurs binaires. Le code de contrôle \cdots surélèvera les points de telle sorte qu'ils soient à la même hauteur que l'opérateur binaire. Ainsi, $a + \cdots + z$ produira a+···+z. Le code de contrôle \ldots inscrira les points sur la ligne de base, et ainsi, $1\1dots n$ produit 1···n.

Ø Exercice 5.6 Composez : x∧(y∨z) = (x∧y)∨(x∧z)

Ø Exercice 5.7 Composez : 2+4+6+...+2n = n(n+1)

Une relation indique une propriété de deux objets mathématiques. Nous savons déjà montrer deux objets à égalité, ou montrer un nombre plus grand ou plus petit qu'un autre nombre (étant donné qu'il

existe des symboles sur la plupart des claviers d'ordinateur). Pour négativer une relation, le code de contrôle \not est placé devant la relation. Voici quelques relations :

Relations

$\leq$	\leq	$\geq$	\geq	$\equiv$	\equiv	$\cong$	\simeq
$\approx$	\approx	$\subset$	\subset	$\subseteq$	\subseteq	$\supset$	\supset
$\supseteq$	\subseteq	$\in$	\in	$\ni$	\ni	$\parallel$	\parralel
$\perp$	\perp						

Ø Exercice 5.8 Composez : $\vec{x} \cdot \vec{y} = 0$ *if and only if* $\vec{x} \perp \vec{y}$

5.2 Fractions

Il existe deux méthodes de composition d'une fraction : elle peut être composée soit sous la forme 1/2 soit sous la forme $\frac{1}{2}$. Le premier exemple est saisi simplement sans séquence de contrôle particulière. Le deuxième exemple utilise le code de contrôle \over et la syntaxe suivante : {numerator> \over <denominator>}. Ainsi, $${a+b \over c+d}.$$ donne

$$\frac{a+b}{c+d}$$

Ø Exercice 5.11 Tapez le texte suivant : $\frac{a+b}{c} \quad \frac{a}{b+c} \quad \frac{1}{a+b+c} \neq \frac{1}{a} + \frac{1}{b} + \frac{1}{c}$

Ø Exercice 5.12 Tapez : Quels sont les points où $\frac{\int}{x}(x,y) = \frac{\int}{x}(x,y) = o$?

5.3 Indices et Exposants

Les indices et les exposants sont particulièrement faciles à saisir en utilisant TEX. Les caractères _ et ^ sont utilisés pour indiquer que le caractère suivant est un indice ou un exposant. Ainsi x^2 donne x^2 et x_2 donne x_2. Pour obtenir plusieurs caractères en indice ou en exposant, il faut les grouper dans des accolades. Ainsi, nous pouvons utiliser x^{21} pour obtenir x^{21} et x_{21} pour obtenir x_{21}. Notez que les exposants et les indices sont composés automatiquement en taille de caractères réduite. La situation est légèrement plus compliquée un deuxième niveau d'indices ou d'exposants. Vous ne pouvez pas utiliser x_2_3 étant donné que ceci pourrait avoir deux interprétations différentes, en l'occurrence x_{2_3} ou ${x_2}_3$; ceci produit deux résultats différents : x_{23} et x_{2_3}, le premier étant la notation mathématique habituelle d'un indice. Par conséquent, vous devez utiliser les accolades complètes pour décrire les niveaux multiples d'indices et d'exposants. Ceci peut être fait pour n'importe quel niveau.

Pour utiliser à la fois des indices et des exposants sur un seul symbole, vous devez utiliser à la fois _ et ^ dans n'importe quel ordre. Ainsi, x_2^1 ou x^1_2 donnera $x^{\frac{1}{2}}$

Ø Exercice 5.13 Composez les formules suivantes : $e^x \quad e^{-x} \quad e^i + 1 = 0 \quad x_0^2$

Ø Exercice 5.14 Composez : $\nabla^2 \int (x, y) = \frac{\partial^2 \int}{x^2} + \frac{\partial^2 \int}{y^2}$

Une méthode similaire est utilisée pour les sommes et les intégrales. La saisie de $\sum_{k=1}^nk^2$ donnera $\sum_{k=1}^n k^2$ et $\int_int^x f(t) donnera $\int_0^x f(t)dt$.

Une autre utilisation de type de saisie concerne les expressions relatives aux limites. Vous pouvez utiliser $\lim_{n\to \infty}{n+1} \over n})^n = e$ pour obtenir : $\lim_{n+\infty}\left(\frac{n+1}{n}\right)^n = e$

.

Ø Exercice 5.15 Composez l'expression suivante : $lim_{x\to 0}(1+x)^n = e$

Ø Exercice 5.16 Composez : le cardinal de $\left(-\infty, \infty\right)$ est $\aleph_1$

Ø Exercice 5.17 Composez : $\lim_{x\to 0} + x^x = 1$

Voici une astuce pour améliorer l'aspect des intégrales : regardez la différence entre $\int_0^x f(t)dt$ et $\int_0^x f(t)\ dt$. Dans le second cas, il y a un espace supplémentaire après

et l'aspect est meilleur; \, a été utilisé pour ajouter l'espace supplémentaire.

Ø Exercice 5.18 Composez l'intégrale suivante : $\int_0^1 3x^2 dx = 1$

5.4 Racines, Carrés et autres

Pour composer une racine carrée, il suffit d'utiliser la formule \sqrt{...}. Ainsi, $\sqrt{x^2+y^2}$ donnera $\sqrt{x^2} + y^2$. Notez que TEX contrôle le positionnement des symboles ainsi que la hauteur et la longueur du radical. Pour écrire des racines cubiques ou autres, les codes de contrôle \root et \of sont nécessaires. Vous obtiendrez $\sqrt[n]{1+x^n}$ à partir de la syntaxe $\root n \of}1+x^n}$.

Une autre possibilité consiste à utiliser le code de contrôle \surd; la saisie de $\surd 2$ produira $\surd 2$.

Ø Exercice 5.19 Composez ce qui suit : $\sqrt{2} \qquad \sqrt{\frac{x+y}{x-y}} \qquad \sqrt[3]{10} \qquad e^{\sqrt{x}}$

Ø Exercice 5.20 Composez : ||x|| = $\sqrt{x \cdot x}$

Ø Exercice 5.21 Composez : $\phi\ (t) = \frac{1}{\sqrt{2\pi}} \int_0^t e^{\frac{-x^2}{2}}\, dx$

5.5 Lignes inférieures et supérieures

Utilisez les syntaxes \overline{...} et \underline{...} pour insérer des lignes au dessus et en dessous des expressions mathématiques. Ainsi, $\overline{x+y}=\overline x + \overline y$ donne $\overline{x+y} = \overline{x} + \overline{y}$. Cependant, remarquez que les lignes au dessus des lettres sont de hauteurs différentes et il est donc nécessaire de prendre quelques précautions. L'utilisation de \overline{\strut x} augmentera la ligne au dessus de x.

Pour souligner un texte non mathématique, utilisez \underbar{...}.

Ø Exercice 5.22 Composez ce qui suit : $\underline{x}\ \overline{y}$ $\overline{\underline{x+y}}$

5.6 Séparateurs, petits et grands

Les séparateurs mathématiques les plus couramment utilisés sont les crochets, les accolades et les parenthèses. Comme nous l'avons vu, ces caractères peuvent être produits en utilisant [ ] \{\} ( ) pour obtenir [ ], { } et ( ). Parfois des séparateurs plus grands augmentent la clarté des expressions mathématiques, comme dans

(ax(b+c))((axb)+c).

Pour obtenir des séparateurs gauches plus grands, les codes de contrôle \bigl, \Bigl, \biggl et \Biggl sont utilisés devant le séparateur; de même, \bigr, \Bigr, \biggr et \Biggr sont utilisés pour les séparateurs droits. Ainsi, $\Bigl [$ and $\Bigr]$ donnera [and].

Voici un tableau comparatif des tailles des séparateurs

Séparateurs de tailles différentes

{	\{	}	\}	(	(	)	)
{	\bigl\{	}	\bigr\}	(	\bigl(	)	\bigr)
{	\Bigl\{	}	\Bigr\}	(	\Bigl(	)	\Bigr)
{	\biggl\{	}	\biggr\}	(	\biggl(	)	\biggr)
{	\Biggl\{	}	\Biggr\}	(	\Biggl(	)	\Biggr)

Si vous le voulez, vous pouvez laisser TEX choisir la taille du séparateur en utilisant les codes de contrôle \left et \right avant les séparateurs. Ainsi, \left [...\right] provoquera l'inclusion du texte entre des crochets de taille appropriée. Attention : chaque utilisation d'un séparateur \left doit avoir un séparateur \right correspondant. (bien que les séparateurs eux-mêmes puissent être différents). Ainsi, $$\left|{a+b \over c+d}right|.$$ donnera :

$$\left|\frac{a+b}{c+d}\right|$$

Séparateurs mathématiques

(	(	(	)	[	[
]	]	{	\{	}	\}
⌊	\lfloor	⌋	\rfloor	⌈	\lceil
⌉	\rceil	⟨	\langle	⟩	\rangle
/	/	\	\backlash	|	|
||	|	↑	\uparrow	⇑	\Uparrow
↓	\downarrow	⇓	\Downarrow	↕	\updownarrow
⇕	\Updownarrow				

Ø Exercice 5.23 Composez $\lceil\lfloor x\rfloor\rceil\leq\lfloor\lceil x\rceil\rfloor$

5.7 Fonctions spéciales

Il existe plusieurs types de fonctions qui apparaissent fréquemment dans les textes mathématiques. Dans une équation du type $sin^2x+cos^2x=1$, les fonctions trigonométriques "sin" et "cos" sont en caractères romans plutôt qu'en caractères italiques. Ceci est une convention mathématique habituelle utilisée pour indiquer que cette définition est descriptive et non le produit de trois variables. Les codes de contrôle \sin et \cos utiliseront automatiquement les caractères appropriés. Voici un tableau de ces fonctions spéciales :

Fonctions mathématiques spéciales

\sin	\cos	\tan	\cot	\sec	\csc	\arcsin	\arccos
\arctan	\sinh	\cosh	\tanh	\coth	\lim	\sup	\inf
\limsup	\liminf	\log	\ln	\lg	\exp	\det	\deg
\dim	\hom	\ker	\\max	\min	\arg	\gcd	\Pr

Ø Exercice 5.24 Composez : $\sin(2\ )=2\sin\quad\cos\quad\cos(2\ )=\cos^2\ -1$

Ø Exercice 5.25 Composez : $\int\csc^2xdx=-\cot x+c\qquad\lim_{\infty\to 0}\frac{\sin}{\quad}=1$

Ø Exercice 5.26 Composez : $\tan(2\ )=\frac{2\tan}{1-\tan^2}$

5.8 Il existe une macro particulière utilisée dans presque tous les documents mathématiques et qui est suffisamment singulière pour nécessiter une explication approfondie. C'est la macro \proclaim. Elle est utilisée pour établir des théorèmes, des corollaires et autres. Le paragraphe suivant la macro \proclaim est divisé en deux parties. La première partie s'étend jusque, et y compris, le premier point suivi d'un espace, et la deuxième partie est constitué du reste du paragraphe. L'idée est que la

première partie pourrait être quelque chose comme "Théorème 1." ou "Corollaire B". La deuxième partie est l'énoncé du théorème ou du corollaire. Voici un exemple :

\proclaim Theorem 1 (H.~G.~Wells). In the country of the blind, the one-eyed man is king.
donnera
Theorem 1 (H.G. Wells). In the country of the blind, the one-eyed man is king.
L'énoncé du théorème peut naturellement contenir des expressions mathématiques.

Ø Exercice 5.27 Composez :
Theorem (Euclid). *There exist an infinite number of primes.*

Ø Exercice 5.28 Composez :
Proposition 1. $\sqrt[n]{\prod_{i=1}^{n} X_i} \leq \frac{1}{n}\sum_{i=1}^{n} X_i$ with equality if and only if $X_1 = \cdots = X_n$.

5.9 Matrices
Les matrices sont composées en utilisant des combinaisons du caractère d'alignement & et le code de contrôle \cr pour indiquer la fin de la ligne. Commencez avec $$\pmatrix{..}$$. Dans l'espace entre les accolades s'insèrent les rangées de la matrices, chaque d'elles se terminant par \cr. Les entrées sont séparées par le &. Par exemple, la saisie de

```
$$\pmatrix{
a & b c d \cr
b & a c+d & c-d \cr
0 & 0 & a+b & a-b \cr
0 & 0 & ab & cd \cr
}.$$
```

donne le document suivant :

$$\begin{pmatrix} a & b & c & d \\ b & a & c+d & c-d \\ 0 & 0 & a+b & a-b \\ 0 & 0 & ab & cd \end{pmatrix}$$

Ø Exercice 5.29 Composez

$$I_4 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Il est possible d'avoir des matrices qui utilisent d'autres paramètres. L'utilisation de \matrix au lieu de \pmatrix omettra les accolades, par conséquent, les séparateurs doivent être explicitement indiqués en utilisant \left et \right. Voici comment vous pouvez changer la matrice de notre premier exemple.

```
$$ \left |
\matrix{
a & b & c & d \cr
b & a c+d & c-d \cr
0 & 0 & a+b & a-b \cr
```

0 & 0 & ab & cd \cr
}
\right | $$

donne le résultat suivant :

$$\left | \begin{matrix} a & b & c & d \\ b & a & c+d & c-d \\ 0 & 0 & a+b & a-b \\ 0 & 0 & ab & cd \end{matrix} \right |$$

Il est même possible d'utiliser \left. et \right. Pour indiquer que le séparateur d'ouverture ou de fermeture est effacée (remarquez l'utilisation du point).

Ø Exercice 5.30 Utilisez une structure de matrice pour composer :

$$|x| = \begin{cases} x & x \geq 0 \\ -x & x \leq 0 \end{cases}$$

Les matrices peuvent également être saisies en ligne mais elles sont plutôt inesthétiques sauf lorsqu'elles ont un nombre limité de rangées.

5.10 Equations affichées

Toutes les expressions mathématiques que nous avons étudiées jusqu'à présent étaient semblables qu'elles soient saisies en ligne ou en titre. Nous allons maintenant étudier les situations qui s'appliquent aux équations en titre seulement.
La première concerne l'alignement des titres multilignes. Ceci est possible avec le caractère d'alignement & et les codes de contrôle \cr et \eqalign. En commençant avec $$\eqalign{...}$$, les équations devant être alignées sont saisies avec \cr à la fin de chacune d'elles. Dans chaque équation il devra y avoir un symbole d'alignement & pour indiquer où l'alignement devrait se produire. En général, l'alignement est effectué sur le signe égal, bien que cela ne soit pas toujours le cas. Par exemple :

$$\eqalign{
a+b &= c+d \cr
x &= w + y +z \cr
m + n + o + p &= q \cr

donnera

$$\begin{aligned} a+b &= c+d \\ x &= w+y+z \\ m+n+o+p &= q \end{aligned}$$

Les équations en titre peuvent être numérotées dans la marge droite ou la marge gauche. Lorsque le code de contrôle \eqno apparaît dans une équation en titre, tout ce qui suit le code de contrôle est inscrit dans la marge droite. Ainsi, $$ x+y=z. \eqno (1)$$ donnera

$$x+y=z \tag{1}$$

Utiliser \leqalignno pour inscrire les numéros des équations sur la gauche.

Enfin, supposons que du texte doive figurer au milieu d'une équation en titre. Ceci peut être réalisé en l'insérant dans une hbox. Nous décrirons les hboxes plus en détail dans le chapitre suivant. Pour l'instant, nous voulons les utiliser pour abréger de manière provisoire* en utilisant des caractères romans ordinaires et également pour permettre l'insertion d'espaces entre les mots (souvenez-vous que tous les espaces sont ignorés lors de la saisie de formules mathématiques). Ainsi, $$X=Y \hbox{ if and only if }x=y. donnera le résultat suivant :

$$X = Y \hbox{ if and only if } x=y.$$

Notez soigneusement les espaces dans la hbox.

Ø Exercice 5.31 Faites les problèmes des pages 180-181 du livre TEX.

CHAPITRE 6
ALIGNEMENT

Il n'est pas inhabituel de vouloir mettre un tableau au milieu d'un texte. Heureusement, TEX facilite cette opération. En fait, il existe deux méthodes distinctes d'alignement d'un texte. La première a recours à l'environnement de tabulation. Cela ressemble au positionnement des taquets de tabulation sur une machine à écrire. Chaque ligne est traitée individuellement en fonction des tabulations existantes, mais avec une souplesse plus grande que celle apportée par une machine à écrire. La deuxième correspond à l'environnement d'alignement horizontal qui crée le tableau complet en fonction d'un modèle préétabli.

6.1 Tabulations

pour aligner un texte en utilisant l'environnement de tabulation, vous devez tout d'abord définir les positions des tabulations en utilisant le code de contrôle \settabs. Cela étant fait, la ligne devant utiliser ces tabulations commence avec le code de contrôle \+ et se termine par \cr (souvenez-vous que l'espacement effectif des lignes sur le fichier d'entrée n'a aucune espèce d'importance).
La méthode la plus simple pour utiliser le code de contrôle \settabs est d'insérer le texte dans des colonnes identiques. L'utilisation de \settabs 4 \columns positionnera des tabulations qui produiront quatre colonnes identiques. La tabulation est ensuite effectuée en utilisant le caractère d'espacement & pour déplacer le texte vers la tabulation suivante. Par exemple,

```
\settab 4 \columns
\+ British Columbia & Alberta & Saskatchewan & Manitoba \cr
\+ Ontario & Quebac & New Brunswick & Nova Scottia \cr
\+ Prince Edward Island & Newfoundland \cr
```

donnera le tableau suivant :

British Columbia	Alberta	Saskatchewan	Manitoba
Ontario	Quebec	New Brunswick	Nova Scottia
	Prince Edward Island	Newfoundland	

Il est à noter qu'il est possible de sauter certaines tabulations, et il n'est pas nécessaire d'utiliser toutes les tabulations d'un coup. Pour que le même tableau utilise cinq colonnes, il suffit d'utiliser \settabs 5 \columns pour redéfinir les taquets de tabulations; alors, les trois mêmes lignes du dernier exemple donneront le résultat suivant :

Dans cet exemple, les colonnes sont plus petites, bien sûr. En fait, il y a deux entrées qui se chevauchent dans la dernière rangée. Ceci est dû au fait que TEX déplacera le texte sur la tabulation suivante même si (contrairement à une machine à écrire) cela entraîne un retour en arrière sur la page.
Il existe une relation intéressante entre les groupes et les tabulations. Les valeurs \settabs ne sont applicables qu'au groupe dans lequel elles sont définies, comme on pouvait s'y attendre. En outre, chaque entrée de tableau est dans un petit groupe qui lui est propre. Ainsi, nous pouvons réaliser une entrée unique en caractères gras, par exemple, en utilisant \bf sans accolade. De plus, pour toute colonne, à l'exception de la dernière, il est possible de centrer l'entrée ou de l'aligner soit sur la gauche soit sur la droite, ou de remplir une colonne avec des lignes ou des points. Chaque entrée a un code de contrôle \hfil implicite à la fin de telle sorte qu'il figurera à la gauche de la colonne par défaut. L'ajout de \hfil au début de l'entrée provoquera le centrage à l'instar du code de contrôle \line. L'ajout de \hfill au début provoquera le déplacement des entrées vers la droite (\hfill agit comme \hfill dans la mesure où il absorbe les espaces inutiles; lorsque que \hfil et \hfill apparaissent, c'est \hfill qui s'impose).

```
\settab 4 \columns
\+ \hfil British Columbia & \hfill Alberta \qqad & \bf Saskatchewan & Manitoba \cr
\+ \hfil Ontario & \hfill Quebec \qqad & \bf New Brunswick & Nova Scottia \cr
\+ \hfil --- & \fill * \qqad & \bf Newfoundland & Prince Edward Island \cr
\+ \dotfill && \hrulefill & \cr
```

British Columbia	Alberta	Saskatchewan	Manitoba
Ontario	Quebec	New Brunswick	Nova Scottia
-	*	Newfoundland	Prince Edward Island

Ø Exercice 6.1 Prenez le tableau des provinces canadiennes ci-dessus et centrez chaque entrée dans sa colonne.

Les positions de tabulation peuvent être différentes de celles de colonnes identiques. La syntaxe générale est d'utiliser une ligne type de la forme \settabs \+ ... & ... & ... \cr. L'espacement entre les caractères d'alignement & détermine la position des tabulations. Par exemple, \settabs \+ \hskip 1 in & \hskip 2 in & \hskip 1.5 in & \cr positionnera la première tabulation à un pouce de la marge gauche, la suivante deux pouces plus loin, et la troisième un pouce et demi plus loin. Il est aussi possible d'utiliser du texte pour déterminer la distance entre les tabulations. Ainsi, par exemple, une autre ligne type est \settabs \+ \quad Province \quad & \quad Population \quad & \quad Area \quad & \cr. La colonne sera alors assez large pour accepter l'en-tête avec une cadrat de chaque côté. Voici un exemple complet :

```
\settabs \+ \quad Year & \quad & \quad Price \quad & \quad Dividend & \cr
\+ \hfill Year      \quad & \quad Price \quad & \quad Dividend \cr
\+ \hfill 1971      \quad & \quad 41--54 \quad & \qquad $2.60 \cr
\+ \hfill 2         \quad & \quad 46--55\quad & \qquad $2.70 \cr
\+ \hfill 3         \quad & \quad 40--53\quad & \qquad $2.87 \cr
\+ \hfill 4         \quad & \quad 46--55\quad & \qquad $2.24 \cr
\+ \hfill 5         \quad & \quad 45--52\quad & \qquad $3.40 \cr
```

Donnera le résultat suivant :

Année	Cours	Dividende
1971	41-54	$2.60
2	41-54	$2.70
3	46-55	$2.87
4	40-53	$3.24
5	45-52	$3.40

Ø Exercice 6.2 Prenez le tableau ci-dessus et déplacez-le vers le centre de la page.

Ø Exercice 6.3 L'une des façons de centrer un bloc de texte, éventuellement de plusieurs lignes de long, est d'utiliser :
$$\vbox{...}$$. Utilisez-la pour centrer le tableau ci-dessus. La ligne \settabs doit-elle être incluse dans \vbox ?

Ø Exercice 6.4 Améliorez votre dernier résultat en insérant une ligne sous l'en-tête de la colonne. Le code de contrôle \hrule insérera une ligne horizontale s'il est introduit entre deux rangées d'un tableau. Recommencez avec le code de contrôle \strut après \+ dans la ligne contenant l'en-tête de la colonne. (\strut rend effectivement l'espacement entre les lignes légèrement plus grand. La taille par défaut peut être modifiée). Remarquez l'espace supplémentaire qui en résulte.

Ø Exercice 6.5 Créez le tableau suivant avec un alignement décimal, c'est à dire avec les séparateurs décimaux les uns au dessus des autres (imaginez le signe dollar comme étant aligné à droite et les chiffres des cents étant alignés à gauche sur le séparateur décimal) :

Plums	1.22
Coffee	1.78
Granola	1.98
Mushrooms	0.63
Kiwi fruit	0.39
Orange juice	1.09
Tuna	1.29
Zucchini	0.64
Grapes	1.69
Smoked beef	0.75
Broccoli	1.09
Total	12.55

Ø Exercice 6.6 Imaginez une méthode pour établir un tableau sommaire en utilisant settabs et avec des entrées du type :
Démarrage \dotfill & fill 1
Des caractères de toutes les tailles \dotfill & \hfill 9.

6.2 Alignement horizontal avec des structures plus complexes.
L'environnement \settabs n'est pas difficile à utiliser, et une fois que la structure est définie, elle peut être utilisée de manière répétitive dans différentes parties du texte qui suit. Ceci a cependant quelques inconvénients. Tout d'abord, la taille de la colonne doit être définie avant même de connaître les entrées. De plus, même si nous voulions simplement que la troisième colonne soit en caractères gras, il fallait le spécifier dans chaque ligne. Ces problèmes peuvent être résolus plus facilement avec l'environnement \halign.

La structure générale de \halign est la suivante :

```
\halign{ <ligne gabarit> \cr
<première ligne d'affichage> \cr
<deuxième ligne d'affichage> \cr
.
.
.
<dernière ligne d'affichage> \cr
```

La ligne gabarit et les lignes d'affichage sont divisées en sections par le symbole d'alignement &. Dans la ligne gabarit, chaque section utilise les codes de contrôle de la même manière que \line{}. Le code de contrôle \hfil, par exemple, peut être utilisé pour afficher au fer à gauche, au fer à droite ou centré. Les polices peuvent être affichées en utilisant \bf, \it, etc. Le texte peut également être saisi dans la ligne gabarit. En outre, le symbole spécial # doit apparaître une fois dans chaque section. Chaque ligne d'affichage est alors définie en remplaçant chaque section de la ligne d'affichage dans la section correspondante de la ligne gabarit à chaque occurrence du symbole #.

Etudiez l'exemple suivant :

```
\halign{\hskip 2 in $#$& \hfil \quad # \hfil & \qqad $#$ & \hfil \quad # \hfil \cr
\alpha & alpha & \delta & \delta \cr
\gamma & gamma & \delta & delta \cr
\epsilon & epsilon & \zeta & zeta & zeta \cr
}
```

La ligne gabarit indique que la première section du texte composé sera toujours définie à deux pouces à l'intérieur vers la gauche et également toujours sous forme de texte mathématique. La deuxième section sera centrée après ajout d'un cadrat d'espace sur la gauche. Les troisième et quatrième sections sont traitées de la même manière. Voici le résultat :

Dans ce cas, la première ligne d'affichage est réalisée en substituant \alpha au premier # dans la ligne de gabarit, alpha pour le second #, \beta pour le troisième et beta pour le quatrième. La ligne complète est alors sauvegardée pour la définition. Ceci se poursuit jusqu'à ce que toutes les lignes soient accumulées, et alors elles sont définies avec chaque colonne aussi large que nécessaire pour accepter toutes les entrées (une implication de ce processus d'accumulation est qu'une table avec des entrées trop nombreuses pourrait provoquer une saturation de la mémoire de TEX; il est donc préférable de ne pas définir des tableaux dépassant une page.
Ainsi, la ligne de gabarit établit la structure pour les entrées du tableau et les lignes d'affichage insèrent les entrées individuelles.

Parfois les lignes horizontales et verticales sont utilisées pour délimiter les entrées dans une table. Pour insérer des lignes horizontales, nous utilisons \hrule, comme nous l'avons fait pour l'environnement \settabs. Cependant, nous ne voulons pas que la règle soit alignée selon le gabarit, alors nous utilisons le code de contrôle \noalign. Ainsi, les lignes horizontales sont insérées en indiquant \noalign{\hrule}; les lignes verticales sont insérées en indiquant \vrule soit dans le gabarit soit dans la ligne d'affichage. Cependant, tout cela n'est pas aussi simple. Supposons que nous prenions notre dernier exemple et que nous changions le gabarit pour obtenir les lignes verticales et insérer également des lignes horizontales.

```
\halign{hskip 2in\vrule\quad $#$\quad & \vrule \hfil\quad # \hfil & \quad \vrule \quad $#$\quad &
\vrule\hfil \quad # \quad \hfil \hfil \vrule \cr
\noalign{\rule}
\alpha & alpha & \beta & beta \cr
\noalign{\rule}
\gamma & gamma & \delta & delta \cr
\noalign{\hrule}
\epsilon & epsilon & \zeta & zeta \cr
\noalign{\hrule}
}
```

Cela ne donne pas exactement ce que vous escomptez

	α	alpha	β	beta
	γ	gamma	δ	delta
	ϵ	epsilon	ξ	zeta

Il y a plusieurs problèmes : le plus évident est l'extension des lignes horizontales, mais également le texte paraît compacté dans les cases. Entre outre, le texte comporte un espace supplémentaire sur la droite au lieu d'être parfaitement centré. Comme pour l'environnement \settabs, les lignes peuvent être plus haute en incluant le code de contrôle \strut dans le gabarit. Un problème supplémentaire peut survenir lorsque la page est définie étant donné que TEX peut écarter légèrement les lignes pour améliorer l'apparence de la page. Ceci laisserait un écart entre les lignes verticales alors nous utilisons le code de contrôle \offinterlineskip dans \halign pour éviter cela. Finalement, nous pouvons éliminer les lignes collées sur la gauche en supprimant \hskip 2 in dans la ligne de gabarit. Pour déplacer le tableau vers la même position, nous utilisons \moveright. Finalement, nous pouvons étudier la façon de centrer le texte en remarquant que l'espace supplémentaire survient dans la ligne de gabarit après le # où les texte est inséré. Ainsi, nous pouvons améliorer notre résultat en utilisant

```
\moveright 2 in
```

```
\vbox{\offiterlineskip
\halign{\strut \vrule \quad $#$\quad &\vrule \hfil \quad #\quad \hfil &\vrule \quad $#$\quad & \vrule
\hfil \quad #\quad \hfil \vrule \cr \noalign{\hrule}
\alpha & alpha & \beta & beta \cr
\noalign{\hrule}
\gamma & gamma a \delta & delta \cr
\noalign{\hrule}
\epsilon & epsilon & \zeta & zeta \cr
\noalign{hrule}
}}
```

pour obtenir :

α	alpha	β	beta
γ	gamma	δ	delta
ϵ	epsilon	ξ	zeta

En règle générale, si nous voulons construire un tableau avec des entrées encadrées qui soit centré sur une page, nous pouvons y parvenir en insérant \vbow dans la ligne \centerline{}. Il existe cependant une astuce permettant d'obtenir un meilleur résultat. Si \vbox est inséré entre des doubles signes dollar, il sera composé sous forme de texte mathématique. Bien sûr, aucun texte mathématique n'est affiché, mais TEX ajoutera un petit espace au dessus et en dessous du tableau nécessaire à l'affichage. Ainsi, un tableau centré avec cet espacement esthétique peut être créé en suivant les quatre étapes suivantes : (1) insérez \vbox entre des signes dollars doubles, (2) insérez \offinterlineskip et \halign dans \vbox, (3) et mettez une ligne gabarit dans \halign avec \strut au début et \vrule entre chaque entrée, (4) chaque rangée du tableau devra être précédée et suivie de \noalign{\hrule}.

La structure à respecter est la suivante :

```
$$\vbox{
\offinterlineskip
\halign{
\strut \vrule # & ...& \vrule # \vrule \cr
\noalign{\hrule}
<contenu 1ère colonne> & < contenu 2ème colonne > & ...& < contenu dernière colonne> \cr
\noalign{\hrule}
...
\noalign{\hrule}
<contenu 1ère colonne> & < contenu 2ème colonne > & ...& < contenu dernière colonne> \cr
\noalign{\hrule}
}
}$$
```

CHAPITRE 7
FONCTIONS PERSONNALISEES

Dans ce chapitre, nous allons créer de nouveaux codes de contrôle. La création de ces nouvelles définitions, appelées également des macros, est l'une des techniques de TEX les plus puissantes. Pour la première application de cette fonction, nous verrons comment une nouvelle définition peut réduire énormément la frappe en remplaçant des chaînes longues par des chaînes courtes.

7.1 Exemples

Le code de contrôle \def est utilisé pour définir de nouveaux codes de contrôle. La façon la plus simple de procéder est d'utiliser \def\newname{...}. Puis lorsque \newname apparaîtra dans votre fichier d'entrée, il sera remplacé par tout ce qui figure entre les accolades dans la définition. Bien sûr, \newname doit satisfaire la convention de syntaxe des séquences de contrôle, c'est à dire qu'il doit être un code de contrôle (toutes les lettres) ou un symbole de contrôle (précisément un caractère non alphabétique). Ainsi, supposez que vous rédigiez un document qui contient la phrase "University of Manitoba" plusieurs fois. Alors \def\un{University of Manitoba} définit une nouvelle séquence de contrôle \um qui peut alors être utilisée à tout moment. La phrase je suis des cours à l'\um prend alors tout son sens. Si le code de contrôle existe, votre nouvelle définition la remplacera (ceci inclut les codes de contrôle définis par TEX, et il faut par conséquent être prudent sur le choix du nom. Cependant, toute définition est limitée au groupe dans lequel elle est définie. Par exemple :

```
\ def\um{University of Manitoba}
I took my first course at the \um.
{
\def\um{University\'e de Montr\'eal}
Then I took my next course at the \um.
}
Finally I took my last course at the \um.
```

donnera :

I took my first course at the University of Manitoba. Then I took my next course at the Université de Montréal. Finally I took my last course at the University of Manitoba.

Souvenez-vous que tous les espaces après un code de contrôle sont ignorés; ceci inclut les mots que vous définissez. Dans l'exemple précédent, tout espace après \um sera ignoré. Cependant, si vus regardez attentivement la fin de la phrase composée selon l'exemple, vous remarquerez un espace supplémentaire. Il peut être éliminé en insérant un % après l'accolade d'ouverture pour transformer le reste de la ligne en commentaire. Ceci s'applique également à la ligne avec les accolades de fermeture.

Dès qu'une nouvelle séquence de contrôle a été définie, elle peut être utilisée dans de nouvelles définitions. Cela permet de créer des lettres type simples. Créons maintenant une simple lettre.

```
\def\letter{
\par \noindent
Dear \name,
 This is a little note tolet you know that your name is \name.
 \hskip 2 in Sincerely yours,
\vskip 2 \baselineskip
\hskip 2 in The NameNoter
\smallskip \hrule
}
```

Cette lettre utilise désormais un code de contrôle \name qui est indéfini pour le moment. Lorsque \letter sera utilisé, la valeur active de \name apparaîtra dans le corps de la lettre. Ainsi :

```
\def\name{Michael Bishop}
\letter
\def\name{Michelle L\'ev\^eque}
\letter
```

produira deux exemplaires de la lettre, chacune ayant le nom approprié, suivie par une règle horizontale :

Cher Michael Bishop,

This is a little note to let you know that your name is Michael Bishop.

Sincerely yours,

The NameNoter

Dear Michelle Lévèque,

This is a little note to let you know that your name is Michelle Lévèque.

Sincerely yours,

The NameNoter

Nous aurions pu insérer n'importe quel texte entre les accolades dans \def\name{...}; il pourrait comporter plusieurs paragraphes et utiliser d'autres séquences de contrôle (bien que dans ce contexte cela puisse paraître curieux). Bien sûr, il est possible d'utiliser \vfill \eject dans la définition de \letter pour éjecter la page lorsque la page est achevée.

Ø Exercice 7.1 Créez un lettre type qui utilise les codes de contrôle \name \address, \city; \state, et \zipcode.

Ø Exercice 7.2 Une liste d'éléments non numérotés est souvent créée en utilisant \item{$\bullet$}. Définissez une macro \bitem qui réalise cette fonction, et utilisez-la pour plusieurs paragraphes. Changez maintenant chaque puce (bullet) par un tiret (notez qu'un simple changement dans la macro ne répercute pas tous les changements nécessaires dans tous les paragraphes).

Ø Exercice 7.3 Imaginons que vous deviez formater plusieurs paragraphes dans un document \hangindent = 30 pt, \hangafter = 4, et \filbreak (ne vous souciez pas du rôle de ces séquences de contrôle). Définissez une seule séquence de contrôle \setpar qui peut alors être placée en face de chaque paragraphe devant être ainsi formaté.

7.2 Insertion de paramètres

Il est possible d'utiliser les macros de diverses manières en permettant de passer outre les paramètres. Cette idée est pratiquement semblable à la ligne gabarit dans l'environnement \halign. Tout d'abord, examinons le cas où il n'y a qu'un paramètre. Dans ce cas-là, une séquence de contrôle est définie par \def\newword#1{...}. Le symbole #1 peut apparaître entre les accolades (plusieurs fois) dans la définition de \newword. Le texte entre les accolades agit comme un gabarit. Lorsque \newword{...}

apparaîtra dans le texte, il utilisera la définition de \newword avec le texte entre les accolades inséré dans les gabarits à chaque occurrence de #1 dans la définition originale. **L'espacement dans la définition originale est ici cruciale; il ne doit y avoir aucun espace avant l'accolade d'ouverture**.

En guise d'illustration, nous pourrions utiliser la lettre type du dernier chapitre de la manière suivante :

```
\def\letter#1{
\par \noindent
Dear #1,

This is a little note to let you know that your name is #1.

\hskip 2 in Sincerely yours,
\vskip \baselineskip
\hskip 2 in The NameNoter
\smallskip \hrule
}
```

Nous pouvons maintenant utiliser

```
\letter{Michael Bishop}
\letter{Michelle L\'ev\^eque
```

pour obtenir

Dear Michael Bishop,

Ceci est un petit message pour vous faire savoir que votre nom est Michael Bishop.

Sincerely yours,

The NameNoter

Dear Michelle Lévèque,

Ceci est un petit message pour vous faire savoir que votre nom est Michelle Lévèque.

Sincerely yours,

The NameNoter

> Définissons maintenant \def\displaytex#1{$$\vbox{\hsize = 12 cm #1}$$} comme nouvelle macro pour afficher le texte. Ensuite \displaytext{...} provoquera l'insertion du texte entre les accolades dans un paragraphe avec une largeur de 12 centimètres et ensuite centré avec de l'espace ajouté au dessus et en dessous. Ce paragraphe a été défini en utilisant la macro \displaytext.

Le paramètre d'une macro ne peut pas dépasser plus d'un paragraphe. Si un nouveau paragraphe est marqué comme faisant partie d'un paramètre, une erreur sera affichée. Ceci est une fonction de

sauvegarde car, dans le cas contraire, l'omission accidentelle d'une accolade de fermeture provoquera l'absorption par TEX du reste du fichier en tant que paramètre.

Ø Exercice 7.4 Définissez une macro \yourgrade de telle sorte que \yourgrade donne le résultat suivant : The grade you received is 89%. Cette macro devra fonctionner avec n'importe quel autre pourcentage, bien sûr.

Il n'est pas vraiment beaucoup plus difficile d'utiliser plus d'un paramètre. La syntaxe utilisée pour définir un nouveau code de contrôle avec deux paramètres est \def\newword\1#2{...}. #1 et #2 peuvent survenir plusieurs fois dans la définition. Lorsque \newword{...}{...} apparaît dans le texte, le texte entre le premier groupe d'accolades remplace #1 dans la définition et le texte entre le second groupe d'accolades remplace #2 dans la définition. Voici un exemple suivi de son résultat.

```
\def\talks#1#2{#1 talks to #2.}
\talks{John}{Jane}
\talks{Jane}{John}
\talks{John}{me}
\talks{She}{Jane}
```

John talks to Jane. Jane talks to John. John talks to me. She talks to Jane.

Ø Exercice 7.6 Ecrivez une macro \frac de telle sorte que \frac{a}{b} produira la fonction $\frac{a}{b}$.

Il est important de ne pas mettre d'espaces entre la première accolade dans la définition. Si tel était le cas, TEX interprétera la définition différemment de la façon décrite ici. Pour plus de deux paramètres, la méthode de définition est semblable. Pour définir un code de contrôle avec trois paramètres, commencez avec \def\newword#1#2#3{...}. Ensuite, #1, #2 et #3 peut survenir entre les accolades. Lorsque \newword{...}{...{...} apparaît dans le texte, le texte entre chaque groupe d'accolades remplace son symbole correspondant dans la définition du code de contrôle. Les paramètres peuvent aller jusqu'à #9.

7.3 Substitution

Il est parfois pratique de pouvoir donner à un code de contrôle un nom de substitution, par exemple, si vous préférez une orthographe différente, vous voudrez peut-être remplacer \centerline par \centreline. Ceci peut être réalisé en utilisant le code de contrôle \let. L'utilisation de \let \centreline = \centerline rend le nouveau (ainsi que l'ancien) code de contrôle disponible. Ceci peut être utilisé avec des formules mathématiques telles que \let \tensor = otimes. Il est alors possible d'utiliser

```
$$ (A \ tensor B) (C tensor D) = AC \tensor BD. $$
```

pour obtenir

$$(A \otimes B)(C \otimes D) = AC \otimes BD.$$

Ø Exercice 7.7 Définissez les séquences de contrôle \11, \cl, et \rl équivalentes à \leftline, \centerline, et \rightline.

Le code de contrôle \let permet aux utilisateurs d'utiliser leurs propres codes de contrôle. Ceci permet un ensemble personnalisé de séquences de contrôle qui peut être utilisé à volonté à la place de celles fournies par TEX.

CHAPITRE 8
L'ERREUR EST HUMAINE

A certains égards, TEX n'atteint pas la perfection absolue. TEX réagira à une entrée invalide en générant un message d'erreur à l'écran si vous l'utiliser interactivement et également dans le journal. Etant donné la complexité de TEX, l'endroit où l'erreur est détectée peut se situer très loin dans le programme et, par conséquent, l'identification détaillée de l'erreur peut être assez longue et fastidieuse. En outre, TEX essaiera de poursuivre malgré les erreurs et indiquera ce qui a été fait au cours de ce processus. Pour cette raison, la lecture de messages d'erreur peut être un peu difficile pour les non initiés. Le principe est de connaître ce qui est important de votre point de vue et ce qui peut être ignoré sans problème. Etudions donc les erreurs classiques et les messages qu'elles génèrent.

8.1 La marque de fin de fichier

La première erreur que nous étudierons est une erreur que tout le monde fait à un moment ou à un autre, en l'occurrence l'omission de \bye à la fin du fichier. Si vous utilisez TEX de manière interactive, une astérisque
*
sera affichée à l'écran et plus rien ne se passera ensuite, étant donné que, n'ayant reçu aucune instruction de fin de traitement, TEX attend une intervention (sur le clavier). Tout ce que vous taperez sera joint à tout ce qui aura pu être saisi dans vos fichiers. La réponse habituelle est \bye<CR> qui arrête le traitement. <CR> est le code utilisé pour terminer une ligne de saisie. Il est appelé retour chariot, entrée, ou simplement la touche de validation de votre terminal. Il est parfois indiqué par une grande flèche à gauche.

8.2 Séquence de contrôle erronée ou inconnue

L'utilisation d'une séquence de contrôle erronée ou de toute autre séquence inconnue de TEX est une erreur classique. Si texte fonctionne en mode batch, un message d'erreur apparaît et le processus se poursuit en ignorant la séquence de contrôle. Lorsque TEX est utilisé interactivement, il est possible de réparer des erreurs (ceci ne change naturellement pas le fichier d'entrée et il faut le faire lorsque le traitement de TEX est terminé). Supposons que nous ayons un fichier d'entrée TEX constitué des deux lignes suivantes :

```
\line{The left side \hfli the right side}
\bye
```

Le code de contrôle devrait bien sûr être \hfil. Voici le message qui devrait être envoyé à votre terminal :

```
! Undefined control sequence.
l.1 \line{ The left side \hfli
                         the right side}
?
```

La première ligne commence avec ! et donne un message d'erreur. Ce message est suivi du numéro de la ligne sur laquelle l'erreur s'est produite et la partie de la ligne qui a été interprétée correctement. La ligne suivante donne la suite de la ligne après l'erreur. A ce point, le point d'interrogation signifie que TEX attend une réponse. Les réponses valides sont les suivantes :

Réponse désirée	Insertion dans TEX	Résultat
Aide (help)	h<CR>	Raison de l'arrêt indiquée sur le terminal
Insertion (Insert)	i<CR>	Ligne suivante insérée dans le fichier d'entrée

		TEX
Sortie (Exit)	x<CR>	Sortie de TEX. Pages terminées dans fichier DVI
Défilement (scroll)	s<CR>	Liste les messages et continue après erreurs mineures
Exécuter (Run)	r<CR>	Liste les messages et continue après tout type d'erreurs
Supprimer (Quiet)	q<CR>	Toutes les indications du terminal supprimées
Continuer (Carry on)	<CR>	TEX continue de la meilleure façon possible

Dans notre dernier exemple, on peut raisonnablement répondre par h<CR> pour obtenir un message d'aide, puis i<CR> pour insérer davantage de texte, (et TEX répond alors par insert>) et finalement \hfil comme code de contrôle valide. Voici le résultat :

```
? h <CR>
Le code de contrôle à la fin de la ligne supérieure
de votre message d'erreur n'a jamais été \def'ed.
S'il est erroné (par exemple '\hobx', tapez 'I' et la
formule correcte (par exemple '\hbox'). Sinon,
continuez et je passerai outre tout ce qui n'a pas
été défini.
? i <CR>
insert>\hfil
[1]
```

Le code final [1] signifie que la première (et unique) page a été achevée et envoyée dans le fichier DVI. Le fichier d'entrée doit bien sûr être corrigé.

8.3 Nom de police incorrect

Un nom de police incorrect est une erreur semblable à celle d'une séquence de contrôle erronée. Le message d'erreur est différent et un peu déroutant au premier abord. Supposez par exemple que la ligne suivante apparaisse dans votre fichier d'entrée :

```
\font\sf = cmss01
```

La formule correcte est cmss10 car les chiffres ont été inversés. Voici les messages d'erreur et d'aide :

```
!  Font \sf=cmss01 not loadable: Metric (TFM) file not found.
<to be read again>
                   /par
\bye ->\par
          \vfill \supereject \end
1.2 \bye
? h <CR>
I wasn't able to read the size data for this font,
so I will ignore the font specification.
[Wizards can fix TFM files using TFtoPL/PLtoTF.]
You might try inserting a different font spec;
e.g. type 'I\font<same font id>=<substitute font name>';
```

The fichier TFM (TEX Metric Font) est un fichier auxiliaire qui est utilisé par TEX. Ce curieux message vous indique simplement que la police que vous avez définie n'existe pas sur votre ordinateur.

8.4 Signes mathématiques non équilibrés

Une autre erreur commune est d'utiliser $ ou $$ pour commencer une expression mathématique et d'oublier ensuite le deuxième $ ou $$ lorsque vous avez terminé. Le texte qui suit est ensuite traité comme du texte mathématique et, pour compliquer les choses, si d'autres expressions mathématiques sont insérées avec un nouveau $ ou $$, elles seront alors traitées comme du texte ordinaire. Il est inutile d'ajouter que de nombreux messages d'erreur peuvent être générés. TEX essaiera de corriger le problème en insérant un nouveau $ ou $$. Dans tous les cas, le problème est corrigé vers la fin du paragraphe étant donné qu'un nouveau paragraphe commencera automatiquement comme du texte ordinaire.

Etudiez les fichiers d'entrée et de sortie corrects suivants :

```
Since $f(x) > 0$, $a<b$, and $f(x)$ is continuous, we know that
$\int_a^b f(x)\, dx >0$.
```

Since f(x) > 0, a < b, since f(x) is conitnuous, we know that $\int_a^b f(x)\, dx > 0$.

Si nous omettons le deuxième signe dollar dans $f(x)$, nous obtenons alors les messages d'erreur et d'aide suivants :

```
! Missing $ inserted.
<inserted text>
                $
<to be read again>
                \intop
\int ->\intop
          \nolimits
1.2 $\int
          _a^b f(x)\ >0$
? h <CR>
I've inserted a begin-math/end-math symbol since I think
you left one out. Proceed, with fingers crossed.
?
```

La ligne commençant par ! indique ce qui a été fait. La ligne commençant par 1.2 nous montre à quel endroit du fichier d'entrée nous étions lorsque l'erreur s'est produite. Comme dans nos autres exemples, la partie de la ligne interprétée avec succès, c'est à dire jusqu'à \int, apparaît sur une ligne, et la suite apparaît sur la ligne suivante. Le texte restant peut sembler quelque peu abscons. Ces messages montrent ce qui se passait au coeur du programme TEX lorsque l'erreur s'est produite. L'utilisateur novice peut les ignorer.

Voici ce que vous obtenez lorsque TEX essaie d'outrepasser l'erreur.

Since f(x) > 0, a < b, since f(x)isconitnuous, weknowthat $\int_a^b f(x)\, dx > 0$.

Une ligne de texte en italique apparaît sans espacement. Ceci est caractéristique d'un texte normal traité comme du texte mathématique. Si vous constatez cela sur votre sortie, c'est que vous avez certainement oublié un $ ou $$.

8.5 Accolades non équilibrées

Il est facile d'oublier ou de déséquilibrer les accolades de fermeture lorsque l'on crée des groupes. Il peut en résulter une erreur relativement mineure mais également un désastre. Supposez, par exemple, que vous ayez {\bf A bold title dans votre texte sans accolade droite de fermeture. Le résultat sera le même que s'il n'y a pas de accolade d'ouverture, c'est à dire que le reste du document sera en gras si aucun autre changement de police n'est effectué. Vous obtiendrez le message suivant à la fin du fichier :

(\end occurred inside a group at level 1)

Si vous avez commis deux fois la même erreur, alors il y aura deux accolades d'ouverture de plus par rapport aux accolades de fermeture, et vous obtiendrez le message suivant :

(\end occurred inside a groupe at level 2)

TEX ne sait pas que l'accolade de fermeture est manquante jusqu'à ce qu'il atteigne la fin du fichier d'entrée. Par conséquent, le message ne vous indique pas l'endroit où vous avez fait la faute. Si le repérage de l'accolade manquante n'est pas évident, il est toujours possible d'insérer \bye au milieu de votre document. TEX traitera alors uniquement la première moitié du texte. En déplaçant \bye à différents endroits, l'erreur peut être localisée. De plus, l'examen de la sortie du document permet de savoir ce qui s'est passé.

Les accolades d'ouverture manquantes sont beaucoup plus faciles à repérer. Voici un fichier d'entrée de deux lignes accompagné des messages d'erreur et d'aide qui en résultent.

```
\bf Here is the start}, but there is the finish.
\bye
```

```
! Too many }'s.
l.1 \bf Here is the start
                        , but there is the finish
? h <CR>
You've closed more groups than you opened
Such booboos are generally harmless, so keep going.
```

Il est tout à fait possible, bien sûr, que la ligne qui est sensée comporter l'accolade gauche manquante ne soit pas sur la ligne où TEX détecte une erreur.

Une accolade manquante dans la définition d'une nouvelle séquence de contrôle peut provoquer une erreur importante. Etant donné qu'une telle définition peut inclure plusieurs paragraphes, elle risque de ne pas être intégrée par la fin d'un paragraphe et donc de provoquer l'accumulation de texte dans la définition inachevée. Il est même possible que TEX arrive à court de mémoire étant donné qu'il absorbe une quantité énorme de texte! Ce type de définition est appelé "définition filante". Voici un exemple de fichier de deux lignes avec une définition filante :

```
\def\newword{the def
\newword
\bye
```

Voici les messages d'erreur et d'aide :

```
Runaway definition ?
->the def
! Forbidden control sequence found while scanning definition of \newword.
<inserted text>
                }
```

<to be read again
\bye
l.3 \bye
? h <CR>
I suspect you have forgotten a '}', causing me
to read past where you wanted me to stop.
I'll try to recover; but is the error is serious,
you'd better type 'E' or 'X' now and fix your file.
? <CR>
No pages of output.

Il y a de toute évidence une erreur grave. Si elle se produit au début d'un fichier (comme dans l'exemple précédent), il n'y aura pas la moindre sortie !

Si une accolade de fermeture est omise lors de l'utilisation d'une macro avec des paramètres, la définition filante sera arrêtée à la fin du paragraphe. Donc, si \def\newword#1{...} a été défini et que vous utilisez \newword{... Sans accolade de fermeture, alors un paragraphe au plus sera altéré.

En résumé, lorsqu'une erreur se produit, notez le numéro de la ligne pour voir le pourcentage de texte traité, et également la ligne commençant avec un point d'exclamation pour obtenir une description de l'erreur. Si l'erreur n'est toujours pas claire, demandez d'autres informations à TEX en tapant h<CR>. Pour les erreurs mineures, TEX peut continuer si vous répétez la commande <CR>.

CHAPITRE 9
NOTIONS APPROFONDIES

Dans ce chapitre, nous allons étudier les fonctions qui permettent d'utiliser TEX avec une plus grande souplesse ou efficacité. En fonction de la longueur croissante des documents, différentes techniques peuvent être utilisées pour rendre leur création plus facile.

9.1 Petits et gros fichiers

TEX peut lire et écrire des fichiers pendant le traitement. Ceci permet d'utiliser des fichiers de taille réduite qui sont plus faciles à manipuler en créant un fichier maître qui lit les fichiers plus petits dans l'ordre approprié. Le présent document, en l'occurrence, est constitué de 10 chapitres et d'une introduction. En outre, il existe des macros qui sont utilisées pour toutes les sections. Les macros peut être insérées dans un fichier appelé, par exemple, Macros.TEX, l'introduction peut être enregistrée dans Intro.tex, et chaque chapitre peut être enregistré dans son propre fichier. Le code de contrôle \input est ensuite utilisé pour lire un fichier. En général, \input filename provoquera la lecture et le traitement immédiats du fichier filename.tex, comme si le texte filename.tex avait fait partie du fichier qui le lisait. Ce fichier peut inclure d'autres fichiers. En fait, il est souvent plus pratique de créer un seul fichier qui lit des parties de taille réduite, par exemple comme suit :

```
\input macros
\input intro
\input sec1
\input sec2
\input sec3
\input sec4
\input sec5
\input sec6
\input sec7
\input sec8
\input sec9
\input sec10
```

Alors que le texte est en cours d'édition intensive, il est possible de traiter seulement certains fichiers en insérant % au début de chaque ligne contenant un fichier devant être ignoré (cette technique est parfois appelée le "commenting out" des fichiers non voulus).

Le code de contrôle \input permet également l'utilisation de macros prédéfinies. Les macros pour une note d'information, par exemple, peuvent être enregistrées dans un fichier appelé memo.tex. Ces macros peuvent définir \hsize; \vzise et d'autres paramètres, et peuvent inscrire la date et l'heure. Une fois que ceci a été défini, toutes les notes peuvent être lancées avec \input memo pour les éditer avec un format commun.

Assurez-vous que vous n'avez pas le code de contrôle \bye dans votre fichier d'entrée; dans le cas contraire, TEX s'arrêtera à cet endroit-là.

Ø Exercice 9.1 Créez un fichier d'entrée TEX permettant de lire un deuxième fichier. Essayez de lire le deuxième fichier deux fois en utilisant deux fois la code de contrôle \iput.

9.2 Ensemble de macros étendu

La conception de macros pouvant être utilisées avec de nombreux types de documents est de toute évidence utile. La plupart des universités, par exemple, ont des critères de format spécifiques et

souvent compliqués pour les thèses. La conception d'un ensemble de macros répondant à toutes ces spécifications pourrait demander beaucoup de temps et être fastidieux. Il est possible d'utiliser la commande \input pour utiliser un tel ensemble de macros, tout comme vous le faites avec vos propres macros. Cependant, TEX propose une fonctionnalité plus efficace pour les grands ensembles de macros.

Un ensemble de macros peut avoir un format spécial qui peut être lu rapidement par TEX. Ce type de format est appelé un fichier de format, et la forme exacte revêt un intérêt purement technique. Ce qui importe c'est qu'il permet TEX de fonctionner avec un grand nombre de séquence de contrôle prédéfinies. Certaines commandes appelées primitives font partie de la définition de TEX.

Ce que nous avons décrit dans ce manuel est parfois appelé TEX simple, et est constitué de primitives TEX accompagnées d'un ensemble de macros dans un fichier de format (qui est habituellement inclus automatiquement dans TEX) appelé palin.fmt. Pour les esprits curieux, tout code de contrôle peut être visualisé en utilisant \show. La commande \show\centerline permettra l'affichage suivant :

```
> \centerline=macro:
#1->\line {hss #1\hss }.
```

sur l'écran et dans le journal. Vous pouvez également utiliser \show avec vos propres macros pour voir si une macro particulière est définie.

De nombreux centres informatique possèdent l'ensemble de macro LATEX. Cet ensemble permet à l'utilisateur de créer un index, une table des matières, et une bibliographie automatiquement. Il permet aussi d'insérer certaines figures graphiques élémentaires telles que des cercles, des ellipses, des lignes et des flèches. LATEX utiliser également des fichiers spéciaux prédéfinis appelés fichiers de style pour définir des paramètres de page spécifiques. De nombreux fichiers de style différents sont disponibles, certains journaux accepteront des documents sur un support magnétique pour un traitement direct s'ils sont préparés à l'aide de LATEX et d'un fichier de style déterminé. Il n'est pas difficile de passer de TEX à LATEX. Un guide d'utilisation du module de macros conçu par l'auteur, Leslie Lamport, est disponible : Latex : A document preparation system.

L'association American Mathematical Society utilise le module AMS-TEX pour ses journaux. Il est disponible auprès de cette association, et les documents peuvent être soumis à leurs journaux sur un support magnétique utilisant AMS-TEX. Un manuel conçu par Michael Spivak, The Joy of Tex, est disponible auprès de l'A.M.S.

D'autres modules existent et il ne fait pas de doute que d'autres seront développés. Ils sont habituellement d'un prix modique et peuvent être très efficaces dans certaines circonstances. Le Groupe d'Utilisateurs Tex annonce la diffusion de nouveaux modules dans ses publications.

9.3 Lignes horizontales et verticales

La création de lignes horizontales et verticales est aisée avec TEX. Lorsque vous tapez du texte, \hrule provoque la fermeture du paragraphe actif, trace une ligne horizontale dont la largeur correspond à la valeur active de \hsize, puis il poursuit jusqu'au paragraphe suivant. Il est possible de spécifier la largeur de \hrule, par exemple \hrule width 5 cm; et vous pouvez également utiliser \vskip ou \bigskip pour insérer un espace au dessus ou en dessous de \hrule. Voici un exemple :

```
\parindent = 0 pt \parskip = 12 pt
Here is the text before the hrule.
\bigskip
\hrule width 3 in
And here is some text after the hrule.
```

qui produit le texte suivant :

Here is the text before the hrule.

_____

And here is the text before the hrule.

En fait, hrule a seulement 3 pouces de large et a également par défaut une hauteur (c'est à dire le décalage de hrule au dessus de la ligne de base sur laquelle les caractères sont définis) de 0,4 points et une profondeur (le décalage de hrule en dessous de la ligne de base sur laquelle les caractères sont définis) de 0 point. Chacun de ces paramètres peut être définis individuellement. Ainsi, si nous changeons le dernier exemple de la manière suivante :

\hrule width 3 in height 2 pt depth 3 pt

nous obtenons

Here is the texte before the hrule

And here is some text after the hrule.

Les trois paramètres width, height, et depth peuvent être indiqués dans n'importe quel ordre.

Vrule peut être défini analogiquement par rapport à hrule en spécifiant les paramètres width, height de depth si nécessaire. Cependant, contrairement à hrule, vrule ne commencera pas automatiquement un nouveau paragraphe lorsqu'il apparaîtra. Par défaut, la valeur de vrule sera de 0,4 point de large, et sera aussi haut que la ligne sur laquelle il est défini. Ainsi,

Here is some text before the vrule
\vrule\
and this follows the rule

donnera

Here is some text before the vrule | and this follows the vrule.

Ø Exercice 9.2 Créez trois lignes horizontales distantes de 15 points, de 3 pouces de longueur, et en retrait d'1 pouce par rapport à la règle gauche.

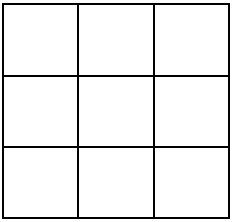
Bien que nous considérions habituellement vrule et hrule comme des lignes horizontales et verticales, elles ne sont pas nécessairement utilisées de cette façon. Par exemple :

\noindent
Name: \vrule height 0 pt depth 0.4 pt width 3 i

donnera

Here is some text before the vrule | and this follows the vrule

Ø Exercice 9.3 Créez la grille suivante (chaque case a 1 cm de côté) :



9.4 Insertion de cadres

Nous avons déjà vu (au cours de notre étude sur la forme des lignes) que vbox et hbox sont des objets qui peuvent être overfull* ou underfull. Dans ce chapitre, nous allons étudier cette question de manière plus détaillée. Ces cases peuvent être empilées ou alignées pour permettre une diversité de positions pour le texte sur la page.

Une case verticale, hbox, est créée en utilisant \hbox{...}. Une fois que le texte entre accolades a été inséré dans une hbox, il est défini et ne pourra plus être scindé (cela signifie que le texte qui doit être mis sur une ligne peut être mis dans une hbox et qu'il y restera ensuite en tant qu'entité distincte). Il est possible de spécifier la taille d'une hbox. Ainsi \hbox to 5cm {texte de la case} produira une hbox de cinq centimètres de large exactement contenant le texte composé "texte de la case". Il est facile d'obtenir une case underfull* ou overfull de cette manière. Une case underfull* peut être évitée en utilisant \hfil pour absorber l'espace supplémentaire. En l'absence de dimension, une hbox de largeur suffisante est créée pour contenir tout le texte.

De même, les cases verticales, vbox sont formées en utilisant \vbox{...}. Ce qui rend ces cases intéressantes c'est que lorsqu'un vbox contient des hbox, ces hbox sont empilées l'une sur l'autre et définies comme entité. De même, une hbox peut contenir des vbox qui seront définies sur une rangée. Supposons que nous voulions prendre trois hbox pour les mettre dans une vbox :

```
\vbox{
    \hbox{Contents of box 1}
    \hbox{Contents of box 2}
    \hbox{Contents of box 3}
    }
```

donne

Contents of box 1
Contents of box 2
Contents of box 3

Supposons maintenant que nous prenions une autre vbox :

```
\vbox{
    \hbox{Contents of box 4}
    \hbox{Contents of box 5}
    }
```

Ces deux vbox peut être insérées dans une hbox; ceci provoquera leur positionnement côte à côte. En d'autres termes :

```
\hbox{
    \vbox{
        \hbox{Contents of box 1}
        \hbox{Contents of box 2}
```

```
        \hbox{Contents of box 3}
        }
    \vbox{
        \hbox{Contents of box 4}
        \hbox{Contents of box 5}
        }
    }
```

donne

Contents of box 1
Contents of box 2 Contents of box 4
Contents of box 3 Contents of box 5

Remarquez que les deux vbox sont alignées de telle sorte que leur base est au même niveau; en outre, un petit espace apparaît au début de chaque ligne et également entre les vbox. En réalité, les raison pour laquelle ces espaces apparaissent est assez subtile. Si une ligne ne se termine pas par un code de contrôle, il y aura toujours un espace entre la dernière entrée sur une ligne et la première sur la ligne suivante. Pour cette raison, l'espace entre les vbox provient de la fin de la ligne contenant l'accolade de fermeture de la première vbox. De même, les espaces au début de la ligne sont provoqués par l'espace suivant l'accolade d'ouverture de la hbox. Ces espaces peuvent être évités en "ignorant" la fin de la ligne, c'est à dire en insérant un % après l'accolade de la première vbox ou l'accolade d'ouverture de la hbox. Si vous essayez de grouper certaines vbox et que vous obtenez de l'espace supplémentaire en oubliant d'ignorer la fin de la ligne, ne vous sentez pas seul. Certains utilisateurs très compétents et expérimentés de TEX ont fait la même chose !

Un espace supplémentaire, par exemple un centimètre, peut être ajouté en insérant \hskip 1 cm entre les vbox. Elles peuvent être alignées de telle sorte qu'elles soient au même niveau en utilisant \vtop au lieu de \vbox. Ces deux changements produisent le résultat suivant dans

Contents of box 1 Contents of box 4
Contents of box 2 Contents of box 5
Contents of box 3

Nous pouvons combiner les vbox, hbox, vrule et hrule pour obtenir un texte encadré. Comment pouvons-nous construire un tel encadrement ? L'une des façons est de prendre le texte à encadrer et de le mettre dans une hbox précédée et suivie d'une vrule. Il suffit ensuite de mettre tout cela dans une vbox avec des hrules au dessus et en dessous d'elle. Cela donne :

```
\vbox{
    \hrule
    \hbox{\vrule Le texte à encadrer\vrule}
    }
```

qui donne le résultat suivant :

Le texte à encadrer

Ceci produit un texte encadré mais sans marge autour et qui paraît comprimé (bien sûr TEX produit ce qu'on lui demande de produire). Nous pouvons améliorer l'espacement en insérant \strut au début de la hbox pour le rendre plus grand et plus ample. Ceci nous donne :

Le texte à encadrer

Ø Exercice 9.4 Pourquoi avons-nous été obligés d'ajouter un espace supplémentaire au dessus et en dessous du texte et non pas avant et après ?

Ø Exercice 9.5 Utilisez la méthode de l'encadrement pour insérer un texte centré dans un cadre qui s'étend de la marge gauche à la marge droite.

Ø Exercice 9.6 En empilant neuf petites cases, créez le carré magique suivant :

6	1	8
7	5	3
2	9	4

Ø Exercice 9.7 Notez que le carré magique de l'exercice précédent a des lignes intérieures qui sont deux fois plus larges que les lignes extérieures. De plus, il y a un petit espace à l'intersection des lignes intérieurs. Modifiez le carré magique de telle sorte que ceci ne se produise pas.

Ø Exercice 9.8 Ecrivez une macro \boxtext#{...} qui prendra le texte entre les accolades et le mettra dans un cadre. Testez votre macro en composant une phrase dont chaque mot est encadré.

Il est facile de déplacer des cadres vers le haut, le bas, la gauche ou la droite de la page. \vbox peut être déplacé vers la droite d'un pouce en utilisant \moveright 1 in \vbox{...}. Pour le déplacer vers la gauche, utilisez \moveleft. De même, \hbox peut être déplacé vers le haut ou le bas en utilisant \raise ou \lower.

Ø Exercice 9.9 Modifiez la macro \boxtext de l'exercice précédent de telle sorte que tout le texte soit aligné (indice : par défaut, la valeur d'un retrait est de 3,5 points). Ceci produira une phrase semblable à la suivante : Je ne suis pas que oserait faire cela car le est assez étrange.

Il est possible de remplir un cadre avec \hrule ou avec des points. L'idée est d'utiliser \hrulefill ou \dotfill dans la hbox.

```
\hbox to 5 in{Getting Started\hrulefill1}
\hbox to 5 in{All characters Great and Small\hrulefill9}
\hbox to 5 in{The shape of Things to come\hrulefill17}
\hbox to 5 in{No Math Anxiety Here!\hrulefill30}
```

donne

Si \hrulefill est remplacé par \dotfill, nous obtenons

Ø Exercice 9.10 Faites apparaître un en-tête encadré en haut de la page et semblable à celui utilisé dans ce manuel.

CHAPITRE 10
LISTE DES CODES DE CONTROLE

Voici une liste est codes de contrôle indiqués dans ce manuel. Si vous voulez davantage de détails sur ces codes, consultez l'index de **The TexBook**.

Symboles de contrôle

\u 4	\! 34	\" 11	\' 11
\, 34	\. 11	\/ 16	\; 34
\% 6	\> 34	\# 10	\$ 6
_ 10	\& 10	\{ 10	\} 10
| 40	\' 11	\~ 10	\^ 10

Codes de contrôle

\AA 12	\aa 12	\acute 36	\AE
\ae 12	\aleph 37	\alpha 35	\angle 37
\approx 37	\arccos 41	\arcsin 41	arctan 41
\arg 41	\ast 36	\b 12	\backlash 37
\bar 36	\baselineskip 22	\beta 35	\bf 16
\bigl 40	\Biggl 40	\biggr 40	\Biggr 40
\bigl 40	\Bigl 40	\bigr 40	\Bigr 40
\bigskip 26	\break 26	\breve 36	\bullet 36
\bye 4	\c 12	\cap 36	\cdot 36
\centerline 26	\centreline 60	\check 36	\chi 35
\circ 35	\columns 48	\cos 41	\cosh 41
\cot 41	\coth 41	\csc 41	\cup 36
\d 12	\ddag 27	\ddot 36	\def 55
\deg 41	\delta 35	\Delta 35	\det 41
\diamond 36	\dim 41	\div 36	\dot 36
\dotfill 49	\dots 14	\downarrow 41	\Downarrow 41
\eject 20	\ell 37	\endinsert 26	\epsilon 35
\eqalign 45	\eqalignno 46	\eqno 46	\equiv 37
\eta 35	\exists 37	\exp 41	\flat 37
\folio 28	\font 16	\footline 28	\footnote 27
\forall 37	\gamma 35	\Gamma 35	\gcd 41
\geq 37	\grave 36	\H 12	\halign 51
\hang 23	\hangafter 23	\hangindent 23	\hat 36
\hbadness 29	\hbox 72	\headline 28	\hfil 27
\hfill 26	\hfuzz 29	\hoffset 20	\hom 41
\hrule 71	\hrulefill 49	\hsize 20	\hskin 27
\hymphenation 30	\i 11	\Im 37	\in 37
\inf 41	\infty 37	\input 68	\int 38
\iota 35	\it 16	\item 24	\itemitem 24
\j 11	\kappa 35	\ker 41	\L 12
\l 12	\lambda 35	\Lambda 35	\langle 41
\lceil 41	\left 44	\leftline 26	\leftskip 23
\leq 37	\leqalignno 46	\leqno 46	\let 61
\lfloor 41	\lg 41	\lim 38	\liminf 41
\limsup 41	\line 26	\ln 41	\log 41
\lower 75	\magnification 21	\magstep 16	\matrix 44