

A n n e x e B

Détecter et résoudre les problèmes

Dans un monde idéal, tous les documents seraient compilés sans problème et donneraient la sortie de haute qualité espérée. Avec de la chance, on n'aura jamais besoin de consulter cette annexe. En revanche, si l'on tombe sur certains problèmes, les informations contenues dans cette annexe pourront nous aider à résoudre les problèmes plus facilement.

Nous commencerons par présenter une liste alphabétique de tous les messages d'erreur, sur lesquels \LaTeX interrompt sa compilation et demande un conseil. *Tous* signifie ici, toutes les erreurs du noyau \LaTeX (le message commence par `LaTeX Error :`), pratiquement tous les messages d'erreur de \TeX (c'est-à-dire ceux produits directement par le moteur sous-jacent) et les erreurs des extensions `amsmath`, `babel`, `docstrip`, `calc`, `color`, `graphics`, `graphicx`, `inputenc`, `fontenc` et `textcomp`. Les erreurs indiquées par les autres extensions – celles qui s'identifient elles-mêmes par

```
! Package <extension> Error : <texte d'erreur>
```

où *<extension>* n'est pas une de celles listées ci-dessus – ne sont pas répertoriées dans cette annexe. Pour ces erreurs, il faut se reporter à la description de l'extension quelque part dans ce livre ou consulter la documentation d'origine de l'extension.

Même si aucune erreur véritable n'interrompt la compilation, on peut voir des messages d'avertissement et d'information sur le terminal ou dans le fichier de trace. Ils sont présentés à la section B.2, où l'on trouve tous les messages du noyau \LaTeX , ainsi que les messages de \TeX qui peuvent retenir l'attention, avec une explication sur leurs causes possibles et comment se comporter face à eux.

La dernière section présente des outils qui permettent d'analyser des problèmes au cas où les erreurs et les avertissements ne seraient pas suffisants ou n'existeraient pas. Nous explorerons plusieurs façons d'afficher les définitions de commandes et les valeurs de registres, puis nous jetterons un œil sur les diagnostics et les résolutions des problèmes de coupures de paragraphes. Nous terminerons par une description de l'extension `trace`, qui permet de suivre l'exécution de commandes au cas où ses propres définitions ou celles d'autres personnes provoquent des résultats inattendus.

Une part du matériel de cette annexe peut être considéré comme du \TeX de bas niveau, des points qui n'ont jamais été décrits dans un livre sur \LaTeX à la connaissance des auteurs. Il s'agit cependant d'informations souvent importantes. Diriger le lecteur vers des livres tels que le *TeXbook* n'est pas d'une grande aide, puisque la plupart des conseils donnés dans ces ouvrages concernent Plain \TeX et ne sont pas applicables à \LaTeX , ou bien produisent des erreurs subtiles lorsqu'ils sont utilisés. Nous essaierons donc de nous limiter aux informations pertinentes sur le moteur \TeX sous-jacent lors d'une utilisation dans un contexte \LaTeX .

B.1 Messages d'erreur

Lorsque \LaTeX s'interrompt pour afficher un message d'erreur, il montre également un numéro de ligne qui indique à quel niveau il en est dans le source. Cependant, en raison de considérations de gestion de mémoire dans la conception même de \TeX , il ne montre pas directement à quel fichier ce numéro de ligne appartient. Pour des documents simples, ce n'est pas un problème mais si un document est découpé en de nombreux fichiers, on doit regarder le terminal ou le fichier de trace avec attention pour identifier le fichier que \LaTeX était en train de traiter lorsque l'erreur est survenue.

*Trouver la ligne
du source contenant
une erreur*

À chaque fois que \LaTeX commence à lire un fichier, il affiche un « (» suivi immédiatement du nom de fichier. Une fois que \LaTeX a fini de lire ce fichier, il affiche un «) » correspondant. De plus, à chaque fois qu'il commence à préparer une page de sortie, il affiche un « [» suivi du numéro de la page en cours. Ainsi, en voyant quelque chose comme

```
(./trial.tex [1] (./ch-1.tex [2] [3] (./table-1.tex [4] [5]) [6]
! Undefined control sequence.
<argument> Un \textss
          {test}
1.235 \section{Un \textss{test}}
                               \label{sec:test}
?
```

on peut en déduire que l'erreur est survenue à l'intérieur d'un argument d'une certaine commande (`<argument>`) et qu'elle a été détectée lorsque \LaTeX examinait le matériel pour la page 7. Dans cet exemple, \LaTeX a lu jusqu'à la ligne 235 du fichier `ch-1.tex`. L'erreur est facile à repérer dans la ligne du source : `\textsf` a été mal orthographié sous la forme `\textss` à l'intérieur de l'argument d'une commande `\section`. Dans certains cas, la relation entre l'erreur et la ligne du source est obscure, voire même inexistante.

Par exemple, en définissant `\renewcommand\thepart{\Alp{part}}`, l'erreur n'apparaît que lorsqu'on utilise la commande `\part` qui exécutera cette définition. Dans ce cas, on obtient :

```
! Undefined control sequence.
\thepart ->\Alp
          {part}
l.167 \part{Test}
```

Dans ce cas, l'erreur véritable ne figure pas à la ligne 167 et sans doute même pas dans le fichier en cours. La commande `\part` appelle simplement la définition fautive de `\thepart`.

ÉTeX détecte parfois une erreur au moment où il prépare une nouvelle page. Comme il s'agit d'une opération asynchrone, la ligne du source indiquée dans le message d'erreur n'a plus aucune valeur. Par conséquent, si l'on ne comprend pas pourquoi une erreur peut être en relation avec une certaine ligne, il se peut que ce soit tout simplement parce qu'il n'y a aucune relation entre elles. Voici un exemple :

```
! Undefined control sequence.
\thepage ->\romen
          {page}
l.33 V
      oici un texte pour remplir la page.
```

On peut obtenir plus d'informations sur une erreur (ou sur la façon dont ÉTeX va la gérer) en répondant `<h>` à la question `?` qui suit le message d'erreur. En l'utilisant avec une erreur T_EX comme celle ci-dessus, on obtient :

```
? h
The control sequence at the end of the top line
of your error message was never \def'ed. If you have
misspelled it (e.g., '\hobx'), type 'I' and the correct
spelling (e.g., 'I\hbox'). Otherwise just continue,
and I'll forget about whatever was undefined.
```

On s'est déjà aperçu du problème avec les conseils du moteur T_EX : il faut les traduire car ils traitent de commandes qui n'appartiennent pas nécessairement au monde ÉTeX (par exemple, pour `\def` il faut lire `\newcommand` ou `\renewcommand`). Avec les erreurs ÉTeX réelles, ce n'est plus le cas, bien que les conseils ne soient pas forcément toujours d'une grande aide.

```
You're in trouble here. Try typing <return> to proceed.
If that doesn't work, type X <return> to quit.
```

Ah ! Merci beaucoup mais nous le savions déjà ! Il s'agit cependant d'un cas extrême et de nombreux messages donnent des conseils plus détaillés.

Afficher la pile des macros partiellement développées

Une autre façon d'obtenir plus d'informations sur une erreur donnée est de modifier le compteur `errorcontextlines` pour lui donner une plus grande valeur. \LaTeX affiche alors la pile des macros en cours de développement :

```

1 ! Undefined control sequence.
2 \thepage ->\romen
3         {page}
4 \@oddfont ->\reset@font \hfil \thepage
5                                     \hfil
6 \@outputpage ...lor \hb@xt@ \textwidth {\@thefoot
7                                     }\color@endbox }}\globa...
8
9 \@opcol ...lumn \@outputdblcol \else \@outputpage
10                                     \fi \global \@mparbotto...
11 <output> ...specialoutput \else \@makecol \@opcol
12                                     \@startcolumn \@whiles...
13 <to be read again>
14         V
15 1.33 V
16         oici un texte pour remplir la page.
```

On lit cela de bas en haut : \LaTeX a vu le V (lignes 15 et 16) mais il veut le lire une nouvelle fois (<to be read again> aux lignes 13 et 14) car il a basculé au niveau de la routine de sortie (output). Il est arrivé jusqu'au développement de la commande `\@opcol` (lignes 11 et 12) qui était appelée par `\@outputpage` (lignes 9 et 10) qui était en train d'exécuter `\@thefoot` (lignes 6 et 7). La ligne 4 est assez curieuse puisqu'elle se réfère à `\@oddfont` plutôt qu'à `\@thefoot`, comme on était en droit de le supposer (`\@thefoot` se développe en `\@oddfont` donc elle est immédiatement développée entièrement et non placée dans la pile des macros partiellement développées). \LaTeX , à l'intérieur de `\@oddfont`, est parvenu jusqu'à l'appel de `\thepage` qui est développé à son tour en `\romen` (lignes 2 et 3) qui, enfin, est marqué comme une commande non définie (ligne 1).

Heureusement, dans la plupart des cas, il suffit d'afficher uniquement le message d'erreur et la ligne du source. C'est pourquoi, pour \LaTeX , la valeur par défaut de `errorcontextlines` est de -1, ce qui signifie qu'aucune ligne de contexte intermédiaire ne doit être affichée.

On peut également avoir des erreurs lorsque \LaTeX traite un fichier intermédiaire utilisé pour transférer des informations entre deux compilations (fichiers `.aux` ou `.toc`, par exemple). Les données de ces fichiers peuvent être corrompues à cause d'une erreur qui survient lors d'une compilation précédente. Même en corrigeant cette erreur dans le source, sa trace dans les fichiers externes peut toujours être présente. Par conséquent, dans certains cas, on doit supprimer ces fichiers avant de compiler à nouveau. Cela dit, ces problèmes disparaissent le plus souvent lors d'une deuxième compilation.

Erreurs persistantes



Les commandes dites *fragiles* sont une source fréquente d'erreurs vicieuses en \LaTeX lorsqu'elles sont utilisées sans être protégées dans des *arguments mouvants*. Techniquement, un argument mouvant est un argument qui est développé en interne par \LaTeX sans être composé directement (par exemple, en utilisant la construction \LaTeX interne `\protected@edef`¹). Plus grossièrement, on peut le voir comme un argument

Erreurs dues aux commandes fragiles



¹Certaines personnes ont entendu dire que la primitive `\edef` existait dans ce but. Il n'est cependant pas conseillé de l'utiliser dans ses propres commandes. Il faut utiliser `\protected@edef` à la place puisque cette commande empêche les commandes fragiles d'être cassées si on les préfixe par un `\protect` !

qui va être placé ailleurs avant d'être composé. C'est, par exemple, le cas des arguments des commandes de sectionnement telles que `\section` (envoyés dans la table des matières), de l'argument de `\caption` (envoyé dans la liste des figures ou des tableaux) et des arguments de `\markboth` et `\markright`.

La meilleure définition, quoique peu utile, d'une commande fragile est qu'il s'agit d'une commande qui produit des erreurs si elle n'est pas précédée d'une commande `\protect` lors d'une utilisation dans un argument mouvant. Les commandes \TeX les plus courantes ont été rendues robustes, et une telle protection n'est plus nécessaire pour celles-ci. Cependant, lorsqu'une commande utilisée dans un argument mouvant produit des erreurs étranges, on peut toujours essayer de la faire précéder d'un `\protect`. Typiquement, les commandes du noyau \TeX avec des arguments optionnels sont fragiles, mais `\sqrt [3] {-1}` est robuste ainsi que *toutes* les commandes avec un argument optionnel définies par l'utilisateur. En revanche, si `\[...]` est fragile sous \TeX standard, cette construction devient robuste une fois l'extension `amsmath` chargée. En d'autres termes, il n'y a aucune règle précise indiquant la « robustesse » d'une commande donnée. Les commandes définies par l'utilisateur avec seulement des arguments obligatoires sont fragiles si elles contiennent des commandes fragiles dans leur définition. Par exemple, la définition

```
\newcommand\frail{\ifthenelse{\value{section}<10 \and
                             \value{subsection}=1}%
                             {\typeout{Oui}}{\typeout{Non}}}
```

est fragile, car l'argument de test de `\ifthenelse` est fragile. Si l'on utilise `\frail` dans l'expression `@` d'un environnement `tabular` (ce qui n'a pas grand sens mais passons),

```
\nonstopmode \begin{tabular}{@{\frail}1} x \end{tabular}
```

on va alors pouvoir admirer pas moins de 134 erreurs avant que \TeX ne veuille finalement redonner la main (dans la liste suivante, la colonne de gauche montre le nombre d'occurrences de chacune des erreurs) :

```
1 ! Argument of \@array has an extra }.
2 ! Argument of \@firstoftwo has an extra }.
1 ! Extra }, or forgotten $.
4 ! Extra }, or forgotten \endgroup.
1 ! LaTeX Error: Illegal character in array arg.
1 ! LaTeX Error: Can be used only in preamble.
51 ! Misplaced \cr.
2 ! Missing # inserted in alignment preamble.
1 ! Missing = inserted for \ifnum.
49 ! Missing \cr inserted.
2 ! Missing control sequence inserted.
2 ! Missing number, treated as zero.
1 ! Missing { inserted.
2 ! Missing } inserted.
1 ! Paragraph ended before \renew@command was complete.
```

```

2 ! Paragraph ended before \reserved@b was complete.
1 ! Paragraph ended before \reserved@c was complete.
2 ! Undefined control sequence.
1 ! Use of \@argtabularcr doesn't match its definition.
7 ! Use of \@array doesn't match its definition.

```

En fait, dans cet exemple, \TeX entre dans une boucle à l'intérieur de laquelle il essaie d'insérer une commande \cr , puis il rejette immédiatement cette idée et répète alors ce processus.

Toutes les erreurs
 \TeX peuvent être
causées par une
commande fragile
dans un argument
mouvant!

Cet exemple montre qu'à chaque fois qu'une erreur \TeX n'ayant pas d'explication simple (par exemple, un nom de commande mal saisi) est rencontrée, elle peut être due à une commande fragile à l'intérieur d'un argument mouvant. On peut donc essayer de la protéger avec un \protect à l'endroit où l'erreur apparaît. Comme cela peut être la raison de chaque erreur \TeX , nous ne répéterons pas cette cause possible pour chacune d'elles (après tout, plus de 60 messages d'erreur \TeX sont présentés ci-dessous).

Erreurs produites
par des clés de
références croisées

Comme on l'a vu à la section A.1.1, il existe quelques restrictions sur les caractères qui peuvent être utilisés dans les clés de référence de \label et \bibitem . En bref, ces clés se comportent parfois comme des arguments mouvants et, selon les caractères utilisés et les extensions chargées, toutes sortes d'erreurs \TeX peuvent survenir. Dans ce cas, la protection avec un \protect *ne fonctionnera pas*. La solution est de n'utiliser que des clés qui se conforment aux restrictions de syntaxe déjà vues.

Liste alphabétique des erreurs \TeX et \LaTeX

Dans la liste suivante, toutes les erreurs \TeX et toutes celles qui proviennent d'extensions sont indiquées avec une référence encadrée à la fin du message d'erreur. Les messages sans cette référence sont des erreurs \LaTeX , le préfixe « \LaTeX Error : » n'étant pas rappelé à chaque fois.

- * Si \LaTeX s'arrête en n'affichant qu'un seul astérisque, cela signifie qu'il a atteint la fin du fichier source sans voir la requête terminant le travail (c'est-à-dire $\end{\document}$ ou \stop) et qu'il attend une entrée à partir du terminal. Même s'il ne s'agit pas d'une erreur en soi, dans la plupart des cas, cela indique que quelque chose s'est particulièrement mal passé. S'il n'y a eu aucune erreur précédente et que le document se termine effectivement par un $\end{\document}$, c'est qu'il y a sans doute eu un environnement `verbatim` non fermé, ce qui a provoqué la lecture de tout le reste du document en mode « verbatim ». Pour trouver la source de ce problème dans un document important, on peut essayer de placer un \end{foo} , qui donnera une erreur « Environment ... ended by... » (indiquant quel environnement \LaTeX est en cours), ou bien qui sera avalé sans réaction, auquel cas on aura la preuve que l'on est effectivement dans un contexte « verbatim ». Dans ce dernier cas, il suffit d'interrompre \LaTeX (en tapant Control-C ou ce que le système d'exploitation demande) et de répondre par « x » à l'erreur d'« interruption » pour terminer la compilation. En réexaminant le résultat à partir de la dernière page, la façon dont le document est composé donne généralement une indication sur l'emplacement où les ennuis commencent.

`'<caractère>' invalid at this point` calc

L'extension `calc` est chargée et la formule utilisée dans une des macros `\setcounter`, `\setlength`, `\addtocounter` ou `\addtolength` utilise une syntaxe incorrecte du point de vue de `calc`. Voir section A.3.1 pour plus de détails.

`<commande> allowed only in math mode` amsmath

Cette commande ou cet environnement ne peut être utilisé qu'en mode mathématique. Il faut alors vérifier attentivement ce qui a été oublié dans le document.

`<nom> undefined`

Cette erreur est déclenchée lorsque `\renewcommand` est utilisé avec un `<nom>` que \TeX ne connaît pas. Soit `<nom>` a été mal saisi, soit il faut utiliser `\newcommand`.

`\< in mid line`

Le `\<` défini à l'intérieur d'un environnement `tabbing` est situé au milieu d'une ligne. Il ne peut être utilisé qu'au début d'une ligne (par exemple, après `\`).

A `<Box> was supposed to be here` TeX

Cette erreur résulte de l'utilisation d'une commande de boîte, telle que `\sbox`, avec un premier argument non valide (c'est-à-dire non déclaré avec `\newsavebox`). Habituellement, on obtient d'abord l'erreur « Missing number, treated as zero », qui indique que \TeX utilise le registre de boîte zéro.

Accent `<commande>` not provided by font family `<nom>` textcomp

L'extension `textcomp` implémente le codage TS1 qui, malheureusement, n'est couvert entièrement que par une minorité de familles de fontes utilisables sous \TeX . Aucun accent n'est imprimé. Voir section 7.5.4 pour savoir comment obtenir une autre représentation de ces accents.

Argument of `<commande>` has an extra `}` TeX

Une accolade droite a été utilisée à la place d'un argument obligatoire de commande (par exemple, `\mbox}`). Les commandes fragiles, lorsqu'elles sont utilisées sans `\protect` dans un argument mouvant, sont souvent cassées sous une forme qui produit cette erreur ou l'une des erreurs « Extra ... » vues plus loin.

Bad `\line` or `\vector` argument

\TeX indique cette erreur lorsqu'on utilise une longueur négative ou une pente non autorisée soit avec `\line`, soit avec `\vector`. Pour le dernier cas, voir le chapitre 10 pour d'autres façons de procéder.

Bad math environment delimiter

Cette erreur est déclenchée lorsqu'une commande `\(` ou `\[` est rencontrée à l'intérieur d'une formule, ou lorsque `\)` ou `\]` est rencontrée en dehors d'une formule mathématique. Vérifier que ces commandes sont convenablement appariées dans le document.

`\begin{<env>}` allowed only in paragraph mode amsmath

Il existe de nombreux emplacements (à l'intérieur du mode horizontal, en mode mathématique, etc.) où il n'y a aucune sens à avoir un hors-texte mathématique. Avec `amsmath`, la totalité de ce hors-texte `<env>` est tout simplement ignoré.

`\begin{<env>}` on input line *<num-ligne>* ended by `\end{<autre-env>}`

ℒ_{TEX} indique cette erreur lorsqu'il voit que l'environnement *<env>* se termine par le code de terminaison de l'environnement *<autre-env>*. La plupart du temps, cette erreur est due à l'oubli du `\end{<env>}`.

Une autre possibilité pour obtenir cette erreur est d'essayer d'utiliser les environnements de type verbatim ou un environnement hors-texte amsmath à l'intérieur de la définition d'un environnement défini par l'utilisateur, ce qui est souvent impossible. Voir section 3.4.3 page 168 pour des solutions utilisant des environnements de type verbatim.

S'il ne s'agit d'aucun des cas précédents et que l'on est absolument certain que tous les environnements sont convenablement emboîtés alors, quelque part entre le début de *<env>* et l'endroit où se situe l'erreur, doit se trouver une commande qui engendre un `\endgroup` sans `\begingroup` antérieur correspondant, ce qui fait croire à ℒ_{TEX} que l'environnement *<env>* prend fin à cet endroit. Pour trouver l'origine du problème, on peut déplacer la commande de fin d'environnement, en la rapprochant du départ de celui-ci, jusqu'à ce que le problème disparaisse.

`\begin{split}` won't work here amsmath

Soit cet environnement `split` ne se situe pas à l'intérieur d'une équation, soit on doit utiliser `aligned`.

Can be used only in preamble

ℒ_{TEX} a rencontré une commande ou un environnement qui doit être utilisé uniquement dans une extension ou le préambule (c'est-à-dire avant `\begin{document}`). Cette erreur peut être également due à un second `\begin{document}`.

Cannot be used in preamble

Certaines commandes (`\nocite` par exemple) ne sont permises que dans le corps du document (c'est-à-dire après `\begin{document}`). Il faut alors déplacer la déclaration à cet emplacement.

Cannot define Unicode char value < 00A0 inputenc

Les valeurs strictement inférieures à "00A0 (160 en décimal) sont soit invalides en tant que valeurs Unicode pour des caractères de texte, soit elles doivent être redéfinies dans ℒ_{TEX}.

Cannot determine size of graphic in *<fichier>* graphics/graphics

La taille explicite de l'image n'a pas été spécifiée et ℒ_{TEX} est incapable de déterminer cette taille directement à partir du *<fichier>* graphique. Normalement, il le fait automatiquement, par exemple avec les fichiers `.eps` en lisant l'information de la boîte englobante. Cependant, en fonction du pilote graphique, il peut être incapable d'extraire cette information à partir d'images bitmap, telles que les fichiers `.jpg`, `.gif` et `.png`.

Cannot include graphics of type : *<ext>* graphics/graphics

On obtient cette erreur lorsqu'on spécifie un type de graphique dans le second argument de `\DeclareGraphicsRule`, ou lorsqu'on utilise le mot-clé `type` de `\includegraphics` non supporté par le pilote graphique chargé.

`\caption outside float`

Une commande `\caption` a été trouvée en dehors d'un environnement flottant tel que `figure` ou `table`. Ce message d'erreur est inactivé par certaines extensions décrites au chapitre 6.

`Command <nom> already defined`

On a essayé de déclarer une commande, un environnement, un nouveau savebox, une longueur ou un compteur avec un `<nom>` qui a déjà une signification pour \LaTeX . Dans ce cas, cette déclaration est ignorée et il faut choisir un autre nom. Cette erreur est également déclenchée lors de l'utilisation de `\newcommand` avec un `<nom>` qui débute par `\end...`, même si `\renewcommand` affirme que ce `<nom>` est inutilisé. Elle survient enfin lorsqu'on essaie de définir un environnement `<nom>` et que les commandes `\<nom>` ou `\end<nom>` ont déjà une définition. Par exemple, on ne peut pas définir un environnement `graf`, car \TeX définit une commande de bas niveau appelée `\endgraf`.

`Command <nom> invalid in math mode`

Il s'agit soit d'un avertissement, soit d'une erreur indiquant que l'on a utilisé une commande en mode mathématique qui ne doit être utilisée qu'en mode texte. Dans le cas du message d'erreur, utiliser `h` pour obtenir plus d'informations.

`Command <nom> not defined as a math alphabet`

Cette erreur survient lorsqu'on essaie d'utiliser `\SetMathAlphabet` sur un `<nom>` qui n'a pas été déclaré précédemment par une commande `\DeclareMathAlphabet` ou `\DeclareSymbolFontAlphabet` pour être défini comme un identificateur d'alphabet mathématique.

`Corrupted NFSS tables`

\LaTeX a essayé une certaine substitution de fontes et a détecté une inconsistance dans ses tables internes. Cette erreur survient lorsqu'une substitution de fontes a été déclenchée et que les règles de substitution contiennent une boucle (c'est-à-dire l'existence de déclarations sub circulaires), ou lorsque les arguments de substitution par défaut pour le codage en cours pointent sur un groupe de formes de fontes inexistant.

`Counter too large`

Cette erreur se produit lorsqu'on essaie d'afficher une valeur de compteur avec `\fnsymbol`, `\alph` ou `\Alph` et que cette valeur est en dehors de l'intervalle autorisé pour la forme d'affichage choisie.

`Dimension too large` TeX

\TeX ne peut gérer que les dimensions dont la taille ne dépasse pas `16383.99998pt` (environ 5,76 mètres) en valeur absolue. Même sur une page énorme, cet intervalle devrait être suffisant.

`\displaybreak cannot be applied here` amsmath

Un environnement extérieur, tel que `split`, `aligned` ou `gathered`, a créé un bloc qui ne peut pas être coupé.

Division by 0 graphics/graphics

On obtient habituellement cette erreur lorsqu'on change l'échelle d'une image ayant une hauteur nulle. Cela peut arriver de façon non intentionnelle, par exemple en spécifiant `[angle=-90,height=3cm]` avec `\includegraphics`. La rotation tourne l'image en lui donnant une hauteur nulle : une valeur difficile à mettre à l'échelle. Dans ce cas, il faut utiliser plutôt `totalheight`.

Double subscript l^AT_EX

Deux indices apparaissent consécutivement (par exemple `xi_2`) et \LaTeX ne sait pas ce que cela signifie : x_{i_2} ou x_{i_2} ? Il faut ajouter des accolades indiquant les indices : `x_{i_2}`.

Double superscript l^AT_EX

\LaTeX a trouvé deux exposants consécutifs. Voir les explications ci-dessus.

Encoding file '*<nom>*' not found fontenc

En demandant le codage *<nom>*, \LaTeX essaie de charger les définitions de ce codage à partir du fichier *<nom>enc.def* (après avoir converti *<nom>* en lettres minuscules). Si ce fichier de codage n'existe pas ou ne peut être trouvé par \LaTeX , on obtient ce message d'erreur.

Encoding scheme *<nom>* unknown

Le schéma de codage *<nom>* qui a été spécifié dans une déclaration ou dans `\fontencoding` n'est pas connu du système. Soit il n'a pas été déclaré en utilisant `\DeclareFontEncoding`, soit le *<nom>* a été mal saisi.

Environment *<nom>* undefined

On obtient cette erreur lorsqu'on utilise `\renewenvironment` ou un nom d'environnement qui n'est pas connu de \LaTeX . Soit le *<nom>* a été mal saisi, soit il faut utiliser `\newenvironment`.

Erroneous nesting of equation structures ; amsmath
trying to recover with 'aligned'

Seules certaines structures hors-texte peuvent être emboîtées ; `aligned` est l'une d'entre elles et le système remplace l'environnement incorrectement emboîté par cette dernière. Ce n'est probablement pas ce qui est voulu et il faudra modifier l'environnement emboîté.

Extra & on this line amsmath

Cette erreur survient uniquement lorsqu'on utilise les anciens environnements d'`amsmath` non décrits dans cet ouvrage. Cette erreur indique un désastre et il faut vérifier avec beaucoup d'attention l'environnement coupable.

Extra alignment tab has been changed to \cr l^AT_EX

Lorsqu'on utilise une structure d'alignement, telle que `tabular`, ou l'un des environnements mathématiques hors-texte (par exemple, `eqnarray` ou `split` de l'extension `amsmath`), chaque ligne est alors divisée en un nombre défini de colonnes séparées par des signes `&`. L'erreur signifie que ces caractères sont trop nombreux, sans doute à cause de l'oubli d'un `\\` indiquant la fin d'une ligne (le nom \TeX pour cette fin de ligne est `\cr`, mais il n'est pas totalement équivalent à `\\`).

Extra `\endgroup` TeX

TeX a vu un `\endgroup` sans `\begingroup` correspondant.

Extra `\or` TeX

TeX a rencontré une primitive `\or` qui n'a pas de condition de bas niveau `\ifcase` correspondante. Le `\or` supplémentaire peut être dû à une mauvaise utilisation de `\ifthenelse`.

Extra `\right` TeX

Cette erreur indique que TeX a trouvé une commande `\right` sans `\left` correspondant dans une formule. Il faut se souvenir que les paires `\left... \right` doivent appartenir à la même « sous-formule ». Elles ne peuvent pas, par exemple, être séparées par un `&` dans un alignement ou apparaître dans des niveaux de groupes différents.

Extra `}, or forgotten $` TeX

Cette erreur est déclenchée lorsque des délimiteurs de formules mathématiques (par exemple `...$, \[...]`) et des accolades de groupes ne sont pas correctement emboîtés. TeX pense qu'il a trouvé un `}` superflu, comme dans `$x}$`, et l'ignore. Alors que dans cet exemple la suppression de l'accolade fermante est le bon choix, ce ne serait plus le cas avec `\mbox\{a}`. Ici, un `\)` fermant a été oublié et le fait de supprimer le `}` va produire des erreurs supplémentaires.

Extra `}, or forgotten \endgroup` TeX

Le groupe courant a commencé avec `\begingroup` (utilisé, par exemple, par `\begin{...}`) mais TeX a trouvé un `}` fermant au lieu du `\endgroup` correspondant. On obtient cette erreur lorsqu'on laisse un `}` isolé à l'intérieur du corps d'un environnement.

File '*<nom>*' not found

TeX a essayé de charger le fichier *<nom>* mais n'a pas pu le trouver, soit parce qu'il n'existe pas, soit parce que le programme TeX sous-jacent regarde au mauvais emplacement. Si le fichier existe et que TeX indique qu'il n'est pas disponible, il est possible que l'installation TeX utilise un mécanisme de hachage pour accélérer l'accès au fichier. Dans ce cas, on doit lancer un programme spécial (par exemple, `mktextsr`).

Cette erreur est déclenchée lorsque des commandes telles que `\input` et `\usepackage` ne peuvent pas trouver le fichier demandé. On peut indiquer un autre nom de fichier en réponse à l'erreur. Si le nouveau nom est spécifié sans extension de nom de fichier, l'ancienne extension est réutilisée si elle est connue de TeX. Si l'on ne veut pas charger de fichier, il faut appuyer sur *<Entrée>*; pour quitter la compilation, il faut saisir `x` ou `X`. Dans certains cas, on reçoit une erreur TeX de bas niveau similaire (`! I can't find file '<nom>'`) qui est un peu plus difficile à quitter. Voir l'entrée page 917.

Si un fichier graphique demandé avec `\includegraphics` est absent, il peut être judicieux de taper `h` pour en apprendre un peu plus sur les extensions testées lorsque le fichier a été recherché.

File ended while scanning *(quelque chose)* TeX

Cette erreur fait partie des erreurs du type « Runaway . . . ». Voir les explications page 925.

Float(s) lost

Un ou plusieurs flottants (par exemple, figure ou table) ou commandes `\marginpar` n'ont pas été composés. La raison la plus fréquente est d'avoir placé un environnement flottant ou une note marginale à l'intérieur d'un `\marginpar`, d'un environnement `minipage`, d'un autre flottant, d'une `\parbox` ou d'une `\footnote`. \LaTeX détecte ce problème très tardivement, lorsqu'il a terminé le document. Cela peut rendre difficile la localisation de l'emplacement fautif. La meilleure solution est de diviser le document en deux de façon répétitive (par exemple en utilisant la primitive `\endinput`) jusqu'à ce que la portion produisant l'erreur soit suffisamment petite pour pouvoir la repérer.

Si un emboîtement incorrect n'est pas la cause principale, on peut avoir rencontré un sérieux problème de codage de l'algorithme des flottants, probablement dû au chargement de certaines extensions.

Font family *(enc)+<famille>* **unknown**

L'erreur résulte d'un essai de déclaration d'un groupe de formes de fontes avec `\DeclareFontShape` sans que la fonte *<famille>* ait été préalablement déclarée disponible dans le codage *<enc>* en utilisant `\DeclareFontFamily`.

Font *<nom>* **not found**

Les tables internes de \LaTeX contiennent une mauvaise information et \LaTeX devient incapable de trouver la fonte externe *<nom>*. Soit cette fonte a été nouvellement installée et \TeX ne peut pas trouver son fichier `.tfm` pour une raison quelconque, soit la déclaration `\DeclareFontShape` s'y référant contient une erreur de frappe.

Font *<nom-interne>=<nom-externe>* **not loadable** : *<raison>* TeX

\TeX est incapable de charger une fonte avec le nom \LaTeX *<nom-interne>* ayant la structure `\<codage>/<famille>/<graisse>/<forme>/<taille>` en notation NFSS¹. Par exemple, il peut s'agir de `\T1/cmr/m/it/10` (Computer Modern medium italique 10 points au codage T1). Cela donne une information intéressante pour savoir sur quelle fonte porte l'erreur, même si l'on ne peut pas en faire grand chose. Pour la *<raison>*, on a les deux possibilités :

Bad metric (TFM) **file** TeX

Le fichier de métrique \TeX de la fonte (c'est-à-dire *<nom-externe>* `.tfm`) est corrompu. L'installation peut disposer de quelques programmes utilitaires permettant de vérifier les fichiers `.tfm` en détail, même si cela exige habituellement l'aide d'un expert.

Metric (TFM) **file not found** TeX

Le fichier de métrique \TeX de la fonte (c'est-à-dire *<nom-externe>* `.tfm`) n'a pas été trouvé. L'installation peut disposer d'une extension (par exemple, `cmbright`)

¹C'est en fait un nom de commande unique mais en raison des barres obliques dans ce nom, on ne peut pas le spécifier directement dans le document.

comme support d'une certaine famille de fontes, mais les fontes correspondantes ne sont pas disponibles ou pas correctement installées.

Font `<nom-interne>=<externe> not loaded : Not enough room left` TeX

TeX ne peut charger qu'un certain nombre de fontes et il n'y a plus de place disponible pour charger `<nom-interne>`. Pour savoir quelles sont les fontes chargées, il faut utiliser l'extension `tracefmt` décrite à la section 7.5.6. Une raison possible pour un chargement excessif de fontes est l'utilisation de tailles de fontes pour lesquelles L^AT_EX doit calculer et charger les fontes mathématiques correspondantes. Voir section 7.10.7 pour plus de détails.

Font shape `<forme-fonte> not found`

Ce message d'erreur survient lorsqu'il y a quelque chose qui ne va vraiment plus du tout avec la déclaration `\DeclareFontShape`, par exemple si elle ne contient aucune spécification de taille. Il faut dans ce cas vérifier l'initialisation pour le groupe de formes de fontes en question.

I can't find file '`<nom>`' TeX

Il s'agit d'une erreur TeX de bas niveau qui a lieu lorsque TeX ne peut pas trouver un fichier qui doit être chargé. Cette erreur ne peut être levée qu'en fournissant à TeX un fichier qu'il est capable de trouver, ou en stoppant la compilation (si le système d'exploitation le permet). Pour passer cette erreur, de nombreuses installations fournissent un fichier `null.tex` qui permet de répondre `null` à cette erreur. L^AT_EX utilise habituellement le message d'erreur « File '`<nom>`' not found » qui permet d'autres actions de la part de l'utilisateur. Cependant, selon le codage de l'extension, on peut obtenir plutôt cette erreur que l'erreur L^AT_EX.

I can't write on file '`<nom>`' TeX

TeX n'est pas capable d'écrire des données dans le fichier `<nom>`. Il est probablement en écriture seule ou il n'y a pas de permission en écriture dans son répertoire. Sur certaines implémentations TeX (comme pour TeX Live), l'erreur peut être précédée d'une ligne telle que :

```
tex: Not writing to /texmf/tex/latex/base/latex.ltx (openout_any = p).
```

Ces installations TeX sont configurées par défaut pour être « paranoïaques » (d'où le « p ») en ce qui concerne l'écriture de fichiers. Elles ne permettent que l'écriture de fichiers situés sous le répertoire courant et interdisent l'écriture de fichiers spécifiés avec un chemin absolu ou dont le nom commence par un point. Ce comportement est modifiable en éditant le fichier `texmf.cnf`.

Illegal character in array arg

On obtient cette erreur lorsque la spécification de colonne dans les environnements `tabular` ou `array`, ou dans une commande `\multicolumn`, contient des caractères non définis comme spécificateurs de colonne. Une cause fréquente est d'utiliser la syntaxe étendue de l'extension `array`, décrite au chapitre 5, en ayant oublié de la charger dans le préambule (par exemple, en ayant copié un tableau à partir d'un autre document).

Illegal parameter number in definition of *commande* TeX

Cette erreur survient lorsqu'une commande ou un environnement (re)défini utilise #<chiffre> dans son texte de remplacement, avec un chiffre strictement supérieur au nombre de paramètres déclarés. Cette erreur peut être provoquée implicitement en raison de commandes de déclarations emboîtées, comme `\newcommand`, en ayant oublié que les commandes intérieures se réfèrent à leurs arguments en doublant les caractères # (voir page 860 pour plus de détails). Une autre cause est de se référer à des arguments d'un environnement dans le second paramètre obligatoire de `\newenvironment` ou `\renewenvironment`.

Illegal unit of measure (pt inserted) TeX

On obtient cette erreur lorsqu'on fait une faute de frappe sur l'unité de longueur ou qu'on l'oublie lors de la spécification de la valeur d'un paramètre de longueur. Voir section A.1.5 pour plus de détails.

Improper argument for math accent : amsmath**Extra braces must be added to prevent wrong output**

La totalité d'une « sous-formule accentuée » doit être entourée d'accolades.

Improper discretionary list TeX

Cette erreur est produite par TeX lorsqu'il rencontre une commande `\discretionary` dont les arguments contiennent autre chose que des caractères, des boîtes ou des blancs fixes après développement.

Improper \hyphenation TeX

Lorsqu'on souhaite spécifier une exception de coupure avec `\hyphenation`, il faut s'assurer que l'argument ne contient que des lettres, et des traits d'union (-) pour indiquer les points de coupure. Le problème est que, par exemple, les caractères accentués sont des glyphes dans certaines fontes (ils sont alors permis), mais que d'autres codages de fontes produisent des constructions complexes exigeant la primitive `\accent`. Par exemple, si l'on utilise le codage T1, `"u` est un glyphe unique. Ainsi,

```
\usepackage[T1]{fontenc} \hyphenation{T"ur-stop-per}
```

est valide. La même exception de coupure utilisée avec le codage par défaut OT1 aurait produit cette erreur. Voir page 463 pour une explication des différences de caractères dans les codages principaux.

Improper \prevdepth TeX

On a utilisé les commandes `\the\prevdepth` ou `\showthe\prevdepth` en dehors du mode vertical, ce qui est interdit. Cette erreur survient également lorsqu'on place un flottant (par exemple, un environnement `figure` ou `table`) à l'intérieur d'un environnement mathématique hors-texte.

Improper \spacefactor TeX

On a utilisé `\the\spacefactor` ou `\showthe\spacefactor` en dehors du mode horizontal, ce qui est interdit.

`\include cannot be nested`

ℒ_{TeX} a rencontré une commande `\include` à l'intérieur d'un fichier chargé avec `\include`. En raison des contraintes d'implémentation, il est impossible d'emboîter ces commandes. On doit remplacer le `\include` intérieur par un `\input`, ou repenser la structure des fichiers afin que toutes les instructions `\include` soient regroupées dans le fichier principal du document.

`Incompatible list can't be unboxed` TeX

On a demandé à $\text{T}_{\text{E}}\text{X}$ de vider une boîte avec un contenu horizontal alors qu'il était en train de construire une liste verticale (ou *vice versa*). Soit il s'agit d'une sérieuse erreur de programmation dans une extension, soit on utilise une commande d'une façon explicitement non prévue. Par exemple, les commandes de l'extension `soul` produisent cette erreur lorsqu'on tente de les emboîter.

`Incomplete <condition>; all text was ignored after line <numéro>` TeX

Une condition $\text{T}_{\text{E}}\text{X}$ de bas niveau ne se termine pas correctement (il n'y a pas de `\fi` correspondant) lorsque $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ atteint la fin du fichier source.

`Infinite glue shrinkage found <quelque part>` TeX

Pour couper un paragraphe en lignes ou la réserve en pages, $\text{T}_{\text{E}}\text{X}$ suppose qu'il n'y a pas de longueur élastique pouvant se compresser arbitrairement puisque cela signifierait que n'importe quelle quantité de matériel pourrait loger dans une seule page. Ainsi, il n'est pas permis d'écrire `\hspace{0pt minus 1fil}` dans un paragraphe ou `\vspace{0pt minus 1fil}` entre deux paragraphes. Le faire provoque cette erreur (*<quelque part>* indique l'emplacement approximatif où le matériel interdit a été trouvé).

`Interruption` TeX

On obtient cette erreur après avoir interrompu la compilation $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ (avec `Control-C` ou ce que le système d'exploitation propose) et cette erreur n'est donc pas surprenante. Pour forcer l'arrêt d'une compilation, on tape `x` suivi de *<Entrée>*. En ne saisissant que *<Entrée>*, la compilation reprend.

`Invalid use of <commande>` amsmath

On a utilisé une commande de `amsmath` à un endroit où elle n'a aucun sens. Il faut revoir la façon d'utiliser cette commande.

`Keyboard character used is undefined in input encoding <enc>` inputenc

Un nombre 8-bits rencontré dans le document ne correspond pas, avec le codage d'entrée *<enc>*, à un objet LICR (voir sections 7.5.2 et 7.11.3). On doit vérifier si le source est vraiment sauvegardé dans le codage indiqué.

`Language definition file <langue>.ldf not found` babel

Lorsque $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ traite la liste d'options de `babel` et lit une option *<langue>* inconnue, il essaie de charger un fichier dont le nom doit être *<langue>.ldf*. Ce message s'affiche lorsque $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ n'arrive pas à trouver ce fichier. L'erreur peut être due à une simple erreur de frappe ou à une sauvegarde du fichier dans un emplacement en dehors du chemin de recherche de $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$.

Limit controls must follow a math operator TeX

On ne peut utiliser `\limits` et `\nolimits` qu'après un opérateur mathématique comme `\sum`. Voir tableau 8.4 page 510 pour une liste des opérateurs usuels.

\LoadClass in package file

`\LoadClass` n'est autorisée que dans les fichiers de classe (voir section A.4).

Lonely \item--perhaps a missing list environment

La commande `\item` n'est autorisée qu'à l'intérieur des structures de liste, et \TeX pense que celle-ci a été trouvée en dehors d'une liste¹.

Math alphabet identifier $\langle id \rangle$ is undefined in math version $\langle nom \rangle$

L'identificateur d'alphabet mathématique $\langle id \rangle$ a été utilisé dans une version mathématique ($\langle nom \rangle$) pour laquelle il n'est pas initialisé. Une déclaration `\SetMathAlphabet` doit être ajoutée au préambule du document pour assigner un groupe de formes de fontes à cet identificateur d'alphabet.

Math version $\langle nom \rangle$ is not defined

Un alphabet mathématique ou une fonte de symboles a été assignée à une version mathématique inconnue. Soit le nom a été mal saisi, soit cette version n'a pas été déclarée (il faut éventuellement charger une extension). Il est également possible que la version mathématique sélectionnée avec `\mathversion` soit inconnue du système.

Misplaced alignment tab character & TeX

\TeX a trouvé un caractère `&` en dehors des environnements d'alignement (`tabular`, `align`, etc.). Pour obtenir un `&`, il faut saisir `\&`. Utiliser un environnement `amsmath`, comme `cases` ou `matrix`, sans avoir chargé cette extension est une cause possible de cette erreur.

Misplaced `\cr` ou Misplaced `\crr` TeX

`\cr` est la commande \TeX de bas niveau qui termine une ligne d'une structure alignée (`\crr` en est une variante). La commande \TeX correspondante est `\\`. \TeX pense qu'il tombe sur cette commande en dehors d'une structure d'alignement.

Misplaced `\noalign` TeX

La primitive \TeX `\noalign` est utilisée en interne pour placer un matériel « non aligné » entre des lignes d'une structure alignée. Cependant, on ne peut l'utiliser qu'immédiatement après une commande qui termine une ligne. Par exemple, on obtient cette erreur lorsqu'on utilise `\hline` en dehors d'un `array` ou d'un `tabular`, ou bien pas immédiatement après un `\\` dans ces environnements.

Misplaced `\omit` TeX

La primitive \TeX `\omit` est utilisée en interne pour modifier les spécifications de colonnes dans une structure alignée (par exemple, pour fusionner des colonnes avec `\multicolumn` dans un `tabular`). La commande `\omit` (et donc les commandes qui l'appellent) n'est permise qu'en tout début d'entrée d'alignement (c'est-à-dire immédiatement après `\\` ou `&`).

¹Contrairement à l'erreur «Something's wrong -- perhaps a missing `\item`, le diagnostic de \TeX est souvent correct dans ce cas.

Missing `\begin{document}`

Cette erreur survient lorsqu'on compose quelque chose dans le préambule du document¹. La plupart du temps, cette erreur est due à une déclaration mal interprétée par \TeX . Elle peut également être produite par un texte sur la même ligne que `\begin{filecontents}`.

Missing control sequence inserted \TeX

On a utilisé `\newcommand` ou `\renewcommand` sans fournir de nom de commande (commençant par une barre oblique inverse) comme premier argument.

Missing `\cr` inserted \TeX

\TeX pense qu'il est temps de terminer une ligne d'un alignement et insère sa commande de bas niveau pour cela. Dans un document \TeX , cette supposition est normalement erronée et la tentative de réparation de \TeX échoue le plus souvent.

Missing delimiter (. inserted) \TeX

Un `\left`, un `\right`, ou l'une des commandes `\big..` n'est pas suivie de son délimiteur. \TeX place un délimiteur vide « . » pour corriger cette erreur (voir section 8.5.3 page 508).

Missing `\endcsname` inserted \TeX

Cette erreur peut survenir en utilisant des commandes dans un nom de compteur ou d'environnement (par exemple, `\newenvironment{Ann'ee}`).

Missing number, treated as zero \TeX

Cette erreur survient lorsque \TeX s'attend à trouver un nombre ou une dimension et trouve autre chose. Par exemple, `\value{page}`, au lieu de `\thepage`, produit cette erreur puisqu'une `\value` isolée fait que \TeX s'attend à une assignation de bas niveau d'un compteur. En général, utiliser un registre de longueur sans fonction appropriée, comme `\setlength`, peut déclencher cette erreur. On obtient également ce message lorsque `\usebox` n'est pas suivi d'un nom de boîte défini par `\newsavebox`, car, en interne, ces noms sont représentés par des nombres.

Missing p-arg in array arg

Un spécificateur de colonne `p` n'est pas suivi d'une expression entre accolades (contenant sa largeur) dans l'argument d'un `tabular`, d'un `array` ou d'un `multicolumn`.

Missing @-exp in array arg

Un spécificateur de colonne `@` n'est pas suivi d'une expression entre accolades (contenant le matériel intercolonne) dans l'argument d'un `tabular`, d'un `array` ou d'un `multicolumn`.

Missing # inserted in alignment preamble \TeX

Un motif d'alignement spécifie la mise en forme des colonnes dans une structure d'alignement. De façon interne, \TeX utilise `#` pour indiquer la partie de la colonne qui reçoit l'entrée. Sous \TeX , cette erreur n'apparaît normalement jamais seule.

¹Composer à l'intérieur des déclarations `\sbox` ou `\savebox` est possible, mais il est plus prudent de placer ce type de déclarations après `\begin{document}`, car certaines extensions peuvent repousser leurs initialisations finales jusqu'à cet emplacement.

Missing = inserted for \ifnum TeX

TeX se plaint qu'une condition de bas niveau `\ifnum` n'est pas suivie de deux nombres séparés par `<`, `=` ou `>`. Cette erreur peut également survenir lorsqu'on oublie l'opérateur de comparaison dans la commande `\ifthenelse` de l'extension `ifthen`.

Missing = inserted for \ifdim TeX

La condition de bas niveau `\ifdim` n'est pas suivie d'un opérateur de comparaison entre deux longueurs.

Missing \$ inserted TeX

TeX a rencontré quelque chose qui n'est autorisé qu'en mode mathématique (par exemple, `\sum`, `\alpha`, `^`) alors qu'il était en mode texte, ou, à l'inverse, quelque chose d'interdit en mode mathématique (par exemple `\par`) alors qu'il traitait une formule. Il a donc inséré un `$` pour passer en mode mathématique ou pour le quitter. Si l'on essaie d'obtenir un trait de soulignement en saisissant `_` au lieu de `_`, TeX composera le reste du paragraphe en mode mathématique, le plus souvent en produisant plusieurs erreurs au cours de ce traitement.

Missing \endgroup inserted TeX

Cette erreur indique qu'une structure de groupe dans le document est mal emboîtée. Les environnements utilisent `\begingroup` et `\endgroup` de façon interne. Lorsqu'on ne peut pas déterminer la cause de cette erreur, on peut utiliser les fonctionnalités `\showgroups` ou `\tracinggroups` de eTeX, comme expliqué à la page 934.

Missing \right. inserted TeX

Une formule contient un `\left` sans le `\right` correspondant. Il convient de se rappeler que les paires de délimiteurs `\left... \right` doivent appartenir à la même « sous-formule ». Elles ne peuvent pas, par exemple, être séparées par un `&` dans un alignement ou apparaître dans des niveaux de groupes différents.

Missing { inserted TeX

TeX pense qu'il manque une accolade ouvrante et en insère une. Cette erreur peut être due à un `}` égaré à l'intérieur d'une entrée de `tabular`.

Missing } inserted TeX

La structure des groupes est fautive pour le document et TeX essaie de la réparer en insérant une accolade fermante. Cette tentative peut réussir ou provoquer d'autres erreurs. Habituellement, le problème devient apparent lorsqu'on regarde la sortie. Lorsqu'on ne peut pas déterminer la cause de cette erreur, on peut utiliser les fonctionnalités `\showgroups` ou `\tracinggroups` de eTeX, comme expliqué à la page 934.

Multiple \label's : label (*étiquette*) will be lost amsmath

On ne peut utiliser qu'une seule commande `\label` par équation à l'intérieur des environnements hors-texte `amsmath`. Il est généralement préférable de toutes les supprimer sauf la dernière, puisque c'est la seule qui aura un effet.

Multiple `\tag` amsmath

On ne peut utiliser qu'une seule commande `\tag` par équation à l'intérieur des environnements hors-texte `amsmath`. Mise à part la première, elles seront toutes ignorées.

No counter '*nom*' defined

Le compteur *nom* référencé par `\setcounter`, `\addtocounter` ou dans l'argument optionnel de `\newcounter` ou de `\newtheorem` est inconnu. Il doit être déclaré au préalable avec `\newcounter`.

No Cyrillic encoding definition files were found babel

Les fichiers de définition de langues pour les « langues cyrilliques » supportées cherchent où tous les codages de fontes connus pour le cyrillique (par exemple T2A, T2B) peuvent être trouvés. Si cette recherche échoue, ce message s'affiche et il faut d'abord installer le support \LaTeX pour le cyrillique.

No declaration for shape *forme-fonte*

Les fonctions de taille `sub` et `ssub` utilisées en argument d'une commande `\DeclareFontShape` se réfèrent à une substitution de forme inconnue de la méthode de sélection de fonte de \LaTeX .

No driver specified color/graphics/graphics

Une des extensions `graphics`, `graphicx` ou `color` a été chargée sans option de périphérique cible. Sur la plupart des installations, cette indication est donnée par les fichiers de configuration `graphics.cfg` et `color.cfg`.

No room for a new *registre* TeX

Les extensions chargées dans le document demandent plus de registres internes (`\count`, `\dimen`, ...) que ce que \TeX peut proposer. Il faut compiler le document avec \eTeX et charger également l'extension `etex`.

No `\title` given

Une classe \LaTeX a exécuté un `\maketitle` sans déclaration `\title`. Seule `\date` est optionnelle lorsqu'on utilise cette commande.

Not a letter TeX

Une exception de coupure a été spécifiée avec `\hyphenation` avec un argument contenant un caractère que \TeX ne considère pas comme une lettre. Par exemple, `\hyphenation{au-jourd'hui}` produit cette erreur puisque ' n'est pas une lettre au sens de \TeX .

Not in outer par mode

On obtient cette erreur lorsqu'un environnement `\marginpar` ou un environnement flottant, tel que `table` ou `figure`, se trouve à l'intérieur d'une commande ou d'un environnement produisant une boîte. Par exemple, on ne peut pas utiliser un `\marginpar` dans une note de bas de page, un flottant, un `tabular`, ou tout autre emplacement analogue (puisque'ils produisent tous des boîtes). On doit déplacer l'objet fautif dans le texte principal.

Number too big TeX

Un nombre a été assigné ou utilisé dans `\setcounter` ou `\addtocounter` tout en étant supérieur au maximum que TeX est capable de gérer (en l'occurrence 2147483647, ou 7FFFFFFF en hexadécimal). Cette erreur peut également survenir lorsqu'on modifie un registre de longueur avec `\setlength` ou `\addtolength`.

OK TeX

Il ne s'agit pas vraiment d'un message d'erreur. On a utilisé une commande de trace de TeX, telle que `\show` ou `\showthe`. Après avoir affiché les données, TeX interrompt la compilation avec ce message pour permettre une interaction sur la ligne de commande (par exemple, pour saisir `i\show...` afin de voir d'autres valeurs). Ce message s'affiche également si `\tracingonline` est strictement positif et que des commandes, qui n'écrivent normalement que dans le fichier de trace, ont été utilisées. Voir le message suivant.

OK (see the transcript file) TeX

Comme pour le message précédent, il ne s'agit pas vraiment d'un message d'erreur. On a utilisé une commande de trace, telle que `\showbox` ou `\showlists`, sans rediriger également l'affichage du résultat sur le terminal.

Old form *<commande>* should be `\begin{env}` amsmath

On a utilisé `cases` ou `matrix` avec leur ancienne syntaxe sous la forme d'une commande qui n'est plus celle d'amsmath. Il faut utiliser la nouvelle syntaxe : celle qui utilise un environnement.

Only one # is allowed per tab TeX

Cette erreur indique un motif d'alignement incorrect. Avec TeX, ce message ne devrait jamais apparaître, sauf si l'on place une commande fragile dans un argument mouvant.

Option clash for package *<nom>*

L'extension *<nom>* a été chargée deux fois avec des ensembles d'options qui entrent en conflit. En appuyant sur H en réponse à cette erreur, TeX montre les options en conflit. Comme TeX ne charge une extension qu'une seule fois¹, la meilleure solution est de spécifier toutes les options à la première occurrence. Si cela n'est pas possible (parce que l'extension est déjà chargée par la classe ou une autre extension), on peut spécifier les options requises en tant qu'options globales au niveau du `\documentclass`. En cas d'urgence, on peut aussi charger une extension avant le `\documentclass` avec `\RequirePackage`. Voir section 2.1.1 pour plus de détails.

Page height already too large

On a utilisé `\enlargethispage` sur une page dont la hauteur est déjà supérieure à 8191.99998pt, c'est-à-dire environ 2,88 mètres. TeX pense qu'il est dangereux d'augmenter la taille de la page de manière aussi importante.

¹La seule exception concerne l'extension `fontenc`, qui peut être chargée autant de fois que nécessaire avec différentes options. Voir section 7.5.3 page 369.

Paragraph ended before *<commande>* was complete 

Comme on l'a vu à la section A.1.2, les commandes définies par `\newcommand*` ou `\renewcommand*` n'acceptent pas de commande `\par` ou de ligne vide dans leurs arguments. Au cas où cela se produit, on obtient un `Runaway` argument ainsi que cette erreur. La *<commande>* indiquée peut ne pas être celle utilisée dans le document. Par exemple, `\emph{. . \par . . }` indiquera la commande `\text@command` dans le message d'erreur (c'est-à-dire la commande interne appelée par `\emph`).

(Please type a command or say 'end') 

On vient de taper *<Entrée>* en réponse à `*`. Voir la toute première entrée page 910.

`\pushtabs` and `\poptabs` don't match

On a utilisé une commande `\poptabs` dans un environnement `tabbing`, mais il n'y avait aucune commande `\pushtabs` précédente.

`\RequirePackage` or `\LoadClass` in Options Section

\LaTeX a trouvé un `\RequirePackage` ou un `\LoadClass` à l'intérieur d'un fichier d'extension ou de classe entre des commandes `\DeclareOption` et `\ProcessOptions`. Le chargement d'extension ou de classe n'est pas permis dans cette partie puisqu'il démolirait la structure de données qui mémorise l'ensemble d'options en cours (voir section A.4). Si l'on veut charger une extension lorsqu'une certaine option est spécifiée, il faut utiliser un drapeau pour indiquer que l'option a été sélectionnée et charger l'extension après que la commande `\ProcessOptions` a fait son travail.

Rotation not supported 

On a effectué une rotation avec `\rotatebox` ou une commande analogue, mais le pilote graphique sélectionné ne gère pas les rotations des objets. \LaTeX laissera l'espacement nécessaire, mais le document imprimé pourra montrer une image à un mauvais emplacement.

Runaway *<quelque chose>* 

\TeX pense qu'il a analysé trop loin alors qu'il recherche la fin du *<quelque chose>*, où *<quelque chose>* peut être argument, définition, `preamble` ou `text`. Outre du code \TeX de bas niveau fautif, le cas le plus fréquent est `argument`. On a, par exemple, oublié de refermer une accolade d'un argument et \TeX recherche la fin de l'argument jusqu'à atteindre la fin du document ou remplir totalement sa mémoire. Des définitions incomplètes avec `\newcommand`, `\newenvironment`, et d'autres déclarations analogues, indiquent également que l'« argument has run away » (l'argument s'emballe). Seules les définitions de bas niveau, avec les primitives de \TeX telles que `\def`, produisent une `Runaway` `definition`.

Un `Runaway` `preamble` signifie qu'une structure d'alignement a des problèmes (cela ne doit pas arriver dans des documents \LaTeX normaux) et `Runaway` `text` indique généralement un problème sur une assignation de registre d'unité lexicale (cela ne doit normalement jamais arriver, à moins d'une sérieuse erreur d'implémentation dans une extension).

Contrairement aux situations avec des messages d'erreur normaux, il n'y a pas de numéro de ligne qui indique où l'erreur a été détectée (puisque \TeX a souvent

atteint la fin du fichier). À la place, on a le début du matériel qui a été absorbé. Par exemple, en cas de définition sans l'accolade fermante, comme

```
\newcommand\foo{bar
\begin{document} Du texte \end{document}
```

on obtient

```
Runaway argument?
{bar \begin {document} Du texte \end {document}
! File ended while scanning use of \@argdef.
<inserted text>
      \par
<*> samplefile.tex

?
```

L'insertion d'un `\par` par \TeX afin de corriger l'erreur n'aide en rien dans ce cas précis, puisque la totalité du document a été avalée. Au lieu de « File ended while... », on peut voir un autre message tel que « Paragraph ended before... ».

Scaling not supported graphics/graphics

On a demandé un changement de taille d'un objet avec `\resizebox` ou une commande analogue, mais le pilote graphique sélectionné ne le gère pas. \LaTeX laisse l'espacement nécessaire, mais le document imprimé montrera l'image avec sa taille inchangée.

Something's wrong--perhaps a missing \item

Ce message d'erreur est produit par une commande `\addvspace` rencontrée en mode horizontal. La supposition « perhaps a missing `\item` » est rarement la bonne. Par exemple, en oubliant de refermer une accolade d'un `\mbox`, comme dans `\mbox{...\section{..}...}`, on va obtenir cette erreur puisque la commande `\section` qui exécute `\addvspace` en interne est maintenant utilisée en mode horizontal.

Il convient d'identifier la commande qui appelle le `\addvspace` provoquant l'erreur, et de comprendre pourquoi elle est mal utilisée. On trouvera une présentation détaillée de la commande `\addvspace` à la page 873.

Sorry, I can't find <format> ... TeX

Si l'on obtient cette erreur, c'est que \LaTeX n'a jamais débuté puisque \TeX ne trouve pas le `<format>` contenant les définitions de base de \LaTeX . Il s'agit d'un problème qui concerne l'installation du système \TeX . Il vaut mieux dans ce cas consulter la documentation de la distribution.

Suggested extra height (<valeur>) dangerously large

Le fait d'utiliser `<valeur>` avec `\enlargethispage` rendrait la page résultante trop haute (plus de 2,88 mètres) aux yeux de \LaTeX .

Symbol font $\langle nom \rangle$ is not defined

On a essayé d'utiliser la fonte de symboles $\langle nom \rangle$ (par exemple, comme argument d'une commande `\DeclareMathSymbol`) sans la déclarer au préalable avec `\DeclareSymbolFont`.

Symbol $\langle commande \rangle$ not provided by font family $\langle nom \rangle$ textcomp

L'extension `textcomp` utilise le codage TS1, qui n'est malheureusement implémenté de façon complète que par une minorité des familles de fontes utilisables avec \LaTeX . L'extension compose le symbole (appelé par $\langle commande \rangle$) en utilisant une famille par défaut mémorisée dans `\textcompstubstdefault`. On peut remplacer l'erreur par un avertissement en chargeant `textcomp` avec l'option `warn`. Voir section 7.5.4 pour plus de détails.

Tab overflow

\LaTeX peut accepter jusqu'à treize positions de tabulation (`(\=)`) à l'intérieur d'un environnement `tabbing`, et on en a utilisé un plus grand nombre. Si tous ne sont pas nécessaires en même temps, on peut résoudre le problème avec `\pushtabs` ou avec des lignes de modèle et `\kill`.

\tag not allowed here amsmath

La commande `\tag` n'est permise qu'au niveau supérieur d'un environnement mathématique hors-texte. Il est généralement préférable de la placer à la fin de l'équation logique à laquelle elle appartient.

TeX capacity exceeded, $\langle explication \rangle$ TeX

\TeX a dépassé ses capacités pour l'un de ces types de mémoires et arrête son travail. Cette erreur est étudiée en détail à la section B.1.1 page 932.

Text line contains an invalid character TeX

Le fichier source contient un caractère étrange, non imprimable, qui est rejeté par \TeX . Cela peut survenir lorsqu'on utilise un éditeur de texte pour créer le fichier qui ne le sauvegarde pas en mode texte.

The attribute $\langle attribut \rangle$ is unknown for language $\langle langue \rangle$ babel

On a essayé d'activer un $\langle attribut \rangle$ d'une $\langle langue \rangle$ qui n'est pas défini dans le fichier de définition de langues pour cette $\langle langue \rangle$. Il faut dans ce cas vérifier la documentation de `babel` sur cette $\langle langue \rangle$.

The character ' $\langle car \rangle$ ' is not a shorthand character in $\langle langue \rangle$ babel

Cette erreur est déclenchée lorsqu'un utilisateur se sert de la commande `\shorthandon` et lui passe un caractère $\langle car \rangle$ qui n'est pas défini comme raccourci pour la $\langle langue \rangle$ en cours. L'instruction est ignorée.

The font size command `\normalsize` is not defined...

Un fichier de classe nécessite une initialisation minimale. La commande `\normalsize` fait partie de ce minimum vital. Voir section A.4.9 page 903 pour plus de détails.

There's no line here to end

Cette erreur est déclenchée lorsqu'une commande `\newline` ou `\\` est trouvée à l'extérieur d'un paragraphe (c'est-à-dire après un `\par` ou une ligne vide). Si l'intention est d'obtenir un espace vertical supplémentaire, il faut utiliser `\vspace` ou l'une des commandes décrites à la page 871.

This may be a LaTeX bug

À la connaissance des auteurs, jusqu'à maintenant, cette erreur n'a jamais indiqué un bogue de \TeX . Elle signifie que \TeX est tellement perdu en raison d'erreurs précédentes qu'il ne sait plus où il en est au niveau de sa structure de données qui gère les flottants. Il vaut mieux dans ce cas interrompre la compilation et corriger les erreurs précédentes.

This NFSS system isn't set up properly

Cette erreur survient lorsque \TeX détecte une faute lors de la vérification des tables de substitutions de fontes au moment du `\begin{document}`. Elle signifie qu'une des déclarations `\DeclareFontSubstitution` ou `\DeclareErrorFont`¹ est corrompue. Ces déclarations nécessitent de pointer sur des formes de fontes valides (déclarées avec `\DeclareFontShape`). On peut taper `h` pour des informations supplémentaires et il faudra en aviser le responsable du système. Si l'on est soi-même le responsable du système, il faudra se reporter à la fin de la section 7.10.5.

Too deeply nested

\TeX standard permet un total de six niveaux d'emboîtement de listes. Ces niveaux peuvent inclure jusqu'à quatre listes de type `itemize` ou de type `enumerate`. Cette erreur signale que le document a dépassé une de ces limites. Le plus probable est d'avoir oublié de fermer certaines listes correctement. Si l'on a vraiment besoin de niveaux supplémentaires, il faut recopier les définitions des environnements `list`, `itemize` et `enumerate` dans une extension privée, et modifier les constantes qui y sont codées en dur.

Too many columns in eqnarray environment

L'environnement `eqnarray` autorise un maximum de trois colonnes (c'est-à-dire deux signes `&` par ligne). Pour des mathématiques sérieuses, on devrait utiliser l'extension `amsmath` décrite au chapitre 8. Elle autorise des structures hors-texte plus complexes.

Too many math alphabets used in version <nom>

On a utilisé un trop grand nombre d'identificateurs d'alphabets mathématiques dans les formules. Si cette erreur survient après avoir ajouté l'extension `bm`, il faut alors définir `\newcommand\bmmax{0}` avant de charger cette extension et essayer de nouveau. Cette définition empêche l'extension `bm` de réaliser une allocation des alphabets mathématiques.

¹La déclaration `\DeclareErrorFont` est utilisée au cours de l'installation et pointe sur une fonte (forme de fonte et taille) qui doit être utilisée lors de la détection d'une erreur. Sa valeur par défaut est Computer Modern Roman 10pt, qui doit être disponible sur n'importe quelle installation \TeX . Voir [116] pour plus d'informations.

Too many unprocessed floats

Les flottants ne pouvant pas être placés immédiatement sont mis en réserve par \LaTeX , ce qui oblige éventuellement les flottants ultérieurs à être placés en réserve à leur tour. \LaTeX peut mettre en attente jusqu'à 18 flottants. Au-delà, on reçoit ce message d'erreur. Cette limite peut être augmentée à 36 en utilisant l'extension `morefloats`, mais si un flottant ne peut pas être placé pour une raison ou pour une autre, cette modification ne fera que retarder l'apparition de l'erreur. Le chapitre 6 indique comment gérer cette situation.

Cette erreur peut également être déclenchée par des commandes `\marginpar` trop nombreuses à l'intérieur d'un paragraphe. Un `\marginpar` utilise temporairement deux emplacements de stockage des flottants en attente tant que le paragraphe en cours n'est pas composé (cela permet donc un maximum de neuf notes marginales par paragraphe, voire moins s'il y a déjà des flottants en attente).

Two `\documentclass` or `\documentstyle` commands

On ne peut utiliser qu'une seule de ces commandes dans un document. Cette erreur indique qu'il y en a plusieurs, peut-être à cause de la combinaison de deux documents séparés.

Two `\LoadClass` commands

Une classe ne peut charger qu'une seule classe au maximum. La section A.4 étudie en détail la façon dont les classes sont construites.

Undefined color $\langle nom \rangle$ color

On a demandé une couleur avec une commande `\color` (ou une commande analogue) de l'extension `color` sans l'avoir préalablement définie avec `\definecolor`. Pour plus de détails, voir [60] ou la documentation de l'extension `color`.

Undefined control sequence TeX

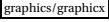
Il s'agit sans doute de l'erreur \LaTeX la plus fréquente, même si elle ressemble à un message d'erreur de \TeX . On a utilisé un nom de commande sans définir celle-ci au préalable. Le plus souvent, on a mal saisi le nom de commande (par exemple, `\bmoX` au lieu de `\mbox`). Dans ce cas, on peut répondre `i\mbox`, ce qui insère le nom correct. Plus tard, on corrigera le fichier source. Il est également possible d'obtenir cette erreur à la suite de l'utilisation d'une commande fragile à l'intérieur d'un argument mouvant.

Undefined font size function $\langle nom \rangle$

Une fonction de taille, utilisée dans `\DeclareFontShape`, a été mal saisie. Il faut vérifier l'entrée ou appeler l'administrateur du système.

Undefined tab position


Cette erreur survient lorsqu'on essaie d'atteindre une position de tabulation dans un environnement `tabbing` avec `\>`, `\+`, `\-` ou `\<`, alors que cette position de tabulation n'a pas été précédemment définie avec `\=`. Soit on a oublié le `\=`, soit on a utilisé `\+` ou `\pushtabs` en se trompant sur la position où l'on voulait effectivement se rendre.

Unknown graphics extension : $\langle ext \rangle$ 

On obtient cette erreur lorsqu'on essaie de charger un fichier graphique (avec l'extension $\langle ext \rangle$), que le pilote graphique ne connaît pas cette extension et qu'il n'y a pas de règle par défaut pour les extensions. Par exemple, le programme `dvips` interprète chaque extension inconnue comme EPS. On n'obtiendra donc jamais cette erreur (mais certainement d'autres) avec ce pilote.

Unknown option ' $\langle option \rangle$ ' for package ' $\langle nom \rangle$ '

On a spécifié une option $\langle option \rangle$ pour l'extension $\langle nom \rangle$ qui n'est pas déclarée par celle-ci. On pourra consulter la documentation de l'extension pour connaître les options disponibles.

Use of $\langle commande \rangle$ doesn't match its definition 

Les définitions de macros de bas niveau, réalisées avec `\def` à la place de `\newcommand` et de ses amis, demandent parfois des délimiteurs d'arguments spéciaux (par exemple, le `(. .)` des commandes d'image). Si $\langle commande \rangle$ est une commande \LaTeX , il faut vérifier sa syntaxe. Sinon, il s'agit le plus souvent d'une erreur parasite due à l'utilisation d'une commande fragile dans un argument mouvant sans `\protect`.

`\usepackage` before `\documentclass`

La déclaration `\usepackage` ne peut être utilisée qu'après avoir chargé la classe principale de document avec `\documentclass`. À l'intérieur d'un fichier de classe, on doit utiliser `\RequirePackage`¹.

UTF-8 string $\langle u8 \rangle$: $\langle séquence-8-bits \rangle$ not set up for \LaTeX use 


Le caractère Unicode indiqué par l'UTF-8 $\langle séquence-8-bits \rangle$ est inconnu de \LaTeX . En supposant qu'il soit disponible avec le codage de fonte utilisé dans le document, on doit le définir avec une déclaration `\DeclareUnicodeCharacter`. Voir section 7.11.3 page 451.

`\verb` ended by end of line

Voici les erreurs les mieux détectées : l'argument d'un `\verb` doit être placé sur une seule ligne. L'erreur signale donc que l'on a oublié le délimiteur final de l'argument ou que l'argument se trouve sur plusieurs lignes dans le source. Dans le cas d'arguments très longs, il peut être utile de les découper en plusieurs commandes `\verb` et, si nécessaire, de masquer les coupures de lignes dans le source avec un signe `%`.

`\verb` illegal in command argument

Sauf dans des situations très spéciales (explicitement indiquées dans ce livre), il n'est pas possible d'utiliser `\verb` (ou `verbatim`) dans les arguments d'autres commandes. Si l'on a besoin de texte verbatim dans ces emplacements, on peut utiliser `\SaveVerb` et `\UseVerb` de l'extension `fancyvrb`, décrite à la section 3.4.3.

You already have nine parameters 

\LaTeX accepte des définitions de commandes ou d'environnements qui ont un maximum de neuf paramètres. L'erreur signale qu'un `\newcommand` ou un `\newenvironment` en spécifie dix ou plus.

¹Il est techniquement possible de charger une extension avant une classe avec `\RequirePackage`, mais on doit l'éviter à moins de savoir ce que l'on fait.

You can't use '*commande*' in *mode* mode TeX

TeX indique que la *commande* n'est pas permise dans le mode *mode*. On a déjà vu des variations spécifiques sur ce thème. Si la *commande* n'a pas été directement utilisée, la cause la plus fréquente de cette erreur est une commande fragile à l'intérieur d'un argument mouvant.

You can't use '\end' in internal vertical mode TeX

C'est l'un des messages d'erreurs de TeX les moins bien compris, puisqu'il se rapporte à la primitive TeX `\end` (terminant une compilation TeX), que TeX redéfinit pour devenir la marque d'une fin d'environnement. L'erreur signifie que la commande `\end{document}` ou la commande `\stop` a été rencontrée pendant que TeX était en train de construire une boîte. Par exemple, `\begin{figure}... \stop` va produire cette erreur.

You can't use 'macro parameter #' in *mode* mode TeX

TeX a trouvé un caractère # dans un endroit où il ne semble pas faire référence à un argument d'une commande. Pour composer ce symbole, il faut utiliser `\#`.

You can't use '\prevdepth' in horizontal mode TeX

La dimension `\prevdepth` ne peut être utilisée qu'en mode vertical (c'est-à-dire entre des paragraphes).

You can't use '\spacefactor' in vertical mode TeX

TeX indique que l'on ne peut utiliser `\spacefactor` que lors de la construction d'une liste horizontale. On obtient cette erreur lorsqu'on utilise la commande `\@` en dehors d'un paragraphe. Comme de nombreux noms de commandes internes débutent par un @, on peut obtenir cette erreur lorsqu'on les utilise dans le préambule du document (par exemple, `\@startsection`), sans avoir entouré ce code avec un `\makeatletter` et un `\makeatother`. Dans ce cas, TeX voit `\@` suivi par les lettres `startsection`, et à la prochaine utilisation de ce code, il exécute donc `\@`, ce qui produit finalement l'erreur.

You haven't defined output directory for '*chemin*' docstrip

Le fichier de configuration `docstrip.cfg` contient une déclaration pour `\BaseDirectory`, mais le *chemin* interne dans le script `DOCSTRIP` n'a pas de traduction en répertoire local. On doit spécifier une telle traduction avec `\DeclareDirectory` ou `\UseTDS`, comme étudié à la section 14.2.3 page 844.

You haven't defined the language *langue* yet babel

De nombreuses commandes d'interfaces de babel vérifient que leur argument est une langue qui a été spécifiée dans la liste d'options lorsque l'extension babel a été chargée. Si la *langue* n'a pas été spécifiée, le traitement est interrompu et ce message d'erreur s'affiche.

You haven't specified a language option babel

On obtient ce message lorsque des langues non connues ont été spécifiées à babel. Cela signifie qu'elles ne sont ni dans la liste d'options de babel, ni dans la liste des options globales (ce qui est, le plus souvent, dû à une erreur de frappe). Il est probable qu'après cette erreur la compilation du document indiquera de nombreuses erreurs supplémentaires.

B.1.1 Mort par dépassement mémoire

Le programme \TeX contient un certain nombre de tables internes de tailles fixées, utilisées pour mémoriser différents types d'informations nécessaires à la compilation. Chaque fois que l'on dépasse la capacité mémoire d'une de ces tables, \TeX abandonne la compilation avec le message d'erreur « `TeX capacity exceeded` ».

Jusqu'au milieu des années 1990, les problèmes de mémoire pouvaient être dus à la taille du document. Dans certains cas, il devenait impossible de traiter le document entièrement¹. De nos jours, de telles limitations ont disparu ou, du moins, sont bien moins sévères. D'une part, une implémentation moyenne de \TeX est déjà équipée de tables internes énormes, et d'autre part, la plupart des implémentations permettent de modifier les tailles des tables via des fichiers de configuration, au lieu de devoir procéder à une recompilation manuelle de \TeX . Dans certains cas, il faudra produire un nouveau format \TeX . Pour plus de détails, on se référera à la documentation du système \TeX de la machine.

Cela dit, tout le monde a vu cette erreur terrifiante un jour ou l'autre, la plupart du temps à cause d'une mauvaise définition de commande. On va présenter ci-dessous quatre candidats réduits à la partie fautive du problème étudié. En réalité, ces problèmes sont généralement cachés à l'intérieur de définitions plus complexes.

```
\newcommand\FAILa{.\FAILa}          \newcommand\FAILb{\FAILb x}
\newcommand\FAILc{\typeout{.}\FAILc} \newcommand\FAILd{.\par\FAILd}
```

En utilisant `\FAILa` comme indiquée ci-dessus, on obtient le résultat suivant (la taille maximale de la mémoire peut être différente) après un court instant :

```
! TeX capacity exceeded, sorry [main memory size=1500001].
\FAILa ->.
      \FAILa
```

La mémoire principale (`main memory`) est la partie de \TeX dans laquelle sont stockées les définitions ainsi que le matériel de la page en cours. En examinant la définition réursive, il est clair qu'elle produit une suite infinie de points. Comme l'algorithme de coupure de paragraphe n'est pas appelé tant que \TeX ne voit pas de commande `\par` ou de ligne vide (afin d'optimiser les coupures de lignes), \TeX attend en vain le moment où il pourra couper ce paragraphe en lignes.

Dépasser la capacité de la mémoire principale à cause d'un nombre trop important de définitions de macros est plutôt rare de nos jours. Cela peut néanmoins arriver (en théorie) lorsque la taille de cette mémoire est petite et que l'on charge de nombreuses extensions, que l'on ait beaucoup de flottants en attente ou que l'on utilise des macros d'extensions² produisant de nouvelles macros à la volée.

¹La première édition de ce livre avait demandé une version spéciale du programme \TeX avec toutes ces tables augmentées d'un facteur 10 et ne pouvait fonctionner que sur de grosses stations de travail UN*X.

²Par exemple `varioref`, qui définit deux étiquettes internes à chaque utilisation de `\vref`, ce qui peut entraîner une consommation de mémoire importante dans des documents volumineux.

Si l'on obtient cette erreur uniquement avec des documents de grande taille et que \LaTeX produit des pages avant d'abandonner, on peut essayer de comprendre pourquoi la mémoire se remplit graduellement (ce qui est suggéré par un problème sur une taille de table), en écrivant l'affectation `\tracingstats=2` dans le préambule du document. \TeX indique alors l'état de la mémoire principale après chaque page, avec une sortie de la forme :

```
[765]
Memory usage before: 4262&161788; after: 1286&157691; still untouched: 1323176
[766]
Memory usage before: 3825&160983; after: 636&156520; still untouched: 1323176
[767]
Memory usage before: 3652&160222; after: 771&156307; still untouched: 1323176
```

Le nombre indiqué à gauche du `&` est la quantité de mémoire utilisée par les grands objets, tels que les boîtes. Le nombre à droite de ce signe est la quantité utilisée par les définitions de macros et les caractères. Ainsi, on peut espérer que ces deux valeurs soient réduites une fois qu'une page est produite (ce qui est montré après le `after` :). Cependant, si les valeurs de la partie droite augmentent légèrement, c'est que quelque chose ajoute de plus de définitions de macros.

En utilisant `\FAILb`, on dépasse les capacités d'une autre table. Ici, la récursion se fait avant que \LaTeX atteigne la fin du développement de la macro et il est donc obligé de mémoriser la partie non traitée du développement :

```
! TeX capacity exceeded, sorry [input stack size=1500].
\FAILb ->\FAILb
      x
```

Aujourd'hui, avec la taille de la pile d'entrée (`input stack`), ce message n'apparaît généralement qu'en cas de récursion fautive comme ci-dessus, qui va faire croître la pile à une vitesse vertigineuse. Dans un document \LaTeX normal, on a rarement besoin d'emboîter des définitions qui feront grandir la pile au-delà de 50 (pour cet ouvrage, la valeur maximale a été de 35).

Que va-t-il se passer lorsqu'on exécute `\FAILc` ou `\FAILd`? Bien que ces deux macros ressemblent beaucoup à `\FAILa`, aucun dépassement de capacité n'est détecté. La seule action de `\FAILc` est de montrer des points, à l'écran et dans le fichier de trace, ce qui ne fait que remplir le disque dur avec un fichier de trace gigantesque. En revanche, `\FAILd` contient un `\par` dans sa définition et il est donc capable de composer des paragraphes (chacun constitué d'un unique point). Le résultat est une production rapide de pages. Une telle expérience sur la machine de d'un des auteurs s'est terminée avec un document contenant 22 279 pages et le message suivant :

```
tex: fwrite: No space left on device
```

Sur une machine personnelle, c'est une simple nuisance, facilement rectifiable. Cependant, sur des systèmes avec des ressources partagées, on doit être prudent lorsqu'on laisse \LaTeX compiler sans surveillance. Une mésaventure est ainsi arrivée à un étudiant

malheureux : il a laissé compiler un tel document sur une station de travail en mode batch, sans limite de temps ou de taille, et s'est vu présenter une facture de plusieurs milliers de dollars pour le temps de calcul.

En principe, on peut arriver à des dépassements de capacité sur plusieurs autres tables internes. On présente ci-dessous la liste complète des tables qui n'ont pas encore été étudiées, avec une explication sur la raison la plus probable du dépassement. On trouvera des informations supplémentaires dans [87, p.300].

buffer size Les caractères sont lus en plaçant chaque ligne du fichier source dans un buffer. Comme la taille par défaut est généralement assez grande, la cause la plus probable d'un dépassement de capacité du buffer est une mauvaise conversion d'un fichier au cours d'un transfert d'un système d'exploitation vers un autre. Un dépassement de capacité du buffer peut également être dû à certains éditeurs du monde PC qui, en interne, placent un paragraphe entier sur une seule ligne, bien que le texte à l'écran soit sur plusieurs lignes.

exception dictionary Le nombre d'exceptions de césures spécifiées par `\hyphenation` est limité par la taille maximale du dictionnaire des exceptions. \TeX spécifie quelques exceptions pour la langue anglaise, et certaines extensions en spécifient pour d'autres langues. Un dépassement de capacité de cette table indique que l'on est en train de réaliser un travail extrêmement minutieux sur les césures!

font memory Les données de métriques de fontes chargées par \TeX sont mémorisées dans la mémoire des fontes. Avec les distributions actuelles, les dépassements de capacité de cette table sont rares. Ils surviennent lorsque \TeX a chargé trop de fontes. Dans la plupart des cas, cela est dû à l'utilisation de nombreuses tailles de fontes, qui oblige \TeX à calculer et à charger les fontes mathématiques pour ces tailles. On peut augmenter la taille de cette table ou se reporter au chapitre 7 pour voir comment réduire le nombre de fontes.

grouping levels Le nombre de groupes (accolades, environnements, délimiteurs mathématiques) non fermés qui détermine la portée des affectations de paramètres, des définitions de macros, etc. est limité. Un débordement indique généralement une erreur de programmation (par exemple, une définition qui ouvre plus de groupes qu'elle n'en ferme). Ce type d'erreur est parfois difficile à identifier. $e\TeX$ ¹ offre une aide précieuse avec la commande `\showgroups`, qui produit une liste des groupes empilés en commençant par le plus intérieur. Par exemple, en plaçant cette commande dans la note de cette page, on obtient :

```
### semi simple group (level 3) entered at line 2405 (\begingroup)
### insert group (level 2) entered at line 2405 (\insert0{)
### semi simple group (level 1) entered at line 2346 (\begingroup)
### bottom level
```

¹Dans les distributions modernes, \TeX utilise automatiquement le programme $e\TeX$. Sur les distributions plus anciennes, on doit réaliser la compilation avec un programme différent (par exemple, `elatex` au lieu de `latex`).

Le groupe de niveau 1 correspond au texte composé dans un environnement `description` (la commande `\begin` appelle `\begingroup` en interne). La commande `\footnote` est implémentée en se fondant sur la primitive TeX `\insert`, qui entre dans le groupe de niveau 2. En réalité, `\footnote` débute un autre groupe afin de s'assurer que les modifications de couleurs resteront locales.

Nous pouvons déduire de cet exemple que les relations entre commandes de haut niveau et groupes internes sont loin d'être évidentes. Cependant, le numéro de ligne qui montre l'emplacement où un groupe a débuté aide beaucoup, puisqu'il n'y a généralement pas de changements de niveaux trop fréquents dans les documents normaux.

Le programme eTeX offre une autre possibilité avec le compteur interne de trace `\tracinggroups`. S'il contient une valeur strictement positive, les entrées et les sorties de groupes sont indiquées dans le fichier de trace. Cette information est aussi affichée à l'écran si `\tracingonline` a une valeur strictement positive.

`hash size` La table de hachage mémorise les noms que TeX connaît. La plupart des extensions contiennent un nombre donné de nouveaux noms de commandes. Chaque commande `\label` ou `\bibitem` du document produit un nouveau nom de commande. Ainsi, les extensions qui utilisent la commande `\label` en interne (comme `varioref`) peuvent grandement contribuer à remplir cette table lorsqu'on travaille sur des documents volumineux.

`number of strings` La table des chaînes mémorise des index qui pointe sur les noms de commandes, de fichiers et de messages d'erreur prédéfinis. Dans certains cas, TeX est capable de libérer de l'espace inutilisé mais, en général, ces chaînes survivent même si elles ne sont utilisées que localement. Une des raisons possibles pour un dépassement de capacité de cette table est l'utilisation de nombreux fichiers dans une application. En effet, chaque ouverture d'un fichier en lecture ou en écriture utilise cette table, même s'il s'agit du même fichier qui est ouvert et fermé de nombreuses fois.

Pour des raisons historiques, TeX a une conception de la gestion des chaînes de caractères assez inhabituelle. Il utilise pour cela plusieurs tables, chacune d'entre elles pouvant provoquer un dépassement de capacité. Par conséquent, si l'on modifie la taille de la table de hachage (`hash size`) pour permettre d'avoir plus de commandes, on doit aussi ajuster la taille de la table des chaînes (`number of strings`) et sans doute également la taille du spouleur (`pool size`) et vice versa.

`parameter stack size` Les paramètres de commandes en train d'être développées, dans le cas de commandes emboîtées, placent des références dans cette pile des paramètres. Par exemple, si une commande à quatre paramètres appelle une commande à cinq paramètres qui, à son tour, appelle une commande à trois paramètres, il y aura finalement douze emplacements utilisés dans cette pile. Lorsque TeX a atteint la fin du texte de remplacement d'une macro, il libère la pile. Avec les implémentations actuelles, il est assez difficile d'obtenir cette erreur en dehors d'une utilisation fautive de définitions récursives avec paramètres. Par exemple :

```
\newcommand\FAIL[3]{\typeout{Got #1, #2 and #3 but \FAIL is a mess}\D0}
```

Voyez-vous le problème ? Puisque `\typeout` contient `\FAIL`, il l'appelle à nouveau, avant que son texte de remplacement soit entièrement traité (en prenant les caractères `i`, `s` et `a` comme arguments). Ainsi, `\DO` n'est jamais utilisée et on obtient :

```
! TeX capacity exceeded, sorry [parameter stack size=1500].
\FAIL #1#2#3->
      \typeout {Got #1, #2 and #3 but \FAIL is a mess}\DO
1.18 \FAIL 123
```

Cela ressemble à l'exemple `\FAILb` à la page 933, mais ici, le premier débordement a lieu sur la pile des paramètres.

`pattern memory` La mémoire des motifs stocke les motifs de césures. Cette table ne peut pas déborder au cours d'un traitement normal de document, puisque ces motifs sont chargés au cours de la production d'un format. Lorsqu'on reçoit cette erreur au cours de ce traitement, il faut réduire le nombre de langues pour lesquelles on charge des motifs de césures dans le format. Actuellement, les motifs à charger sont généralement indiqués dans le fichier `language.dat`.

`pool size` Cette table mémorise les noms de commandes et de fichiers (y compris le chemin complet sur certaines implémentations). La cause la plus fréquente d'un dépassement de capacité est l'utilisation d'un trop grand nombre de fichiers, en particulier s'ils ont des noms très longs. Cela peut être le cas lorsqu'on inclut de nombreux graphiques et qu'on utilise `\graphicspath` pour une recherche des images dans plusieurs répertoires : chaque essai d'ouverture d'un fichier remplit cette table.

`save size` L'ensemble des valeurs à restaurer lorsqu'un groupe prend fin est sauvegardé dans cette pile de sauvegarde. Avec les limites actuelles, il est difficile d'avoir un dépassement de capacité sur celle-ci. La cause la plus fréquente d'erreur est de réaliser des affectations globales *et* locales sur un même objet, ce qui ne peut être réalisé qu'avec des instructions `TEX` de bas niveau, puisque les affectations de `LATEX` sont toujours locales (pour la plupart des objets) ou globales (par exemple, pour les affectations de compteurs).

Pour éviter une augmentation inutile de la pile de sauvegarde, l'environnement `document` est implémenté de façon spéciale¹ afin qu'il ne produise pas de groupe (contrairement aux autres environnements). Sans cela, chaque nouvelle définition empilerait automatiquement une valeur « `undefined` », ce qui serait inutile puisqu'au moment où le groupe prend fin, le traitement s'arrête.

`semantic nest size` À chaque fois qu'une liste d'unités lexicales débute (boîte, formule mathématique, etc.), et qu'une autre liste était en cours de traitement, cette dernière est mise en attente et sauvegardée dans cette pile. Une fois que le travail est fini, `TEX` dépile la liste précédente pour reprendre son travail dessus. Avec une valeur par défaut de plusieurs centaines d'objets, il est très difficile de s'approcher de

¹Un effet de bord est qu'il est impossible d'utiliser `\begin{document}` à l'intérieur d'un autre environnement, puisque la structure de groupe ne serait plus correctement emboîtée.

cette limite avec des documents normaux¹. En cas d'urgence, T_EX permet d'utiliser `\showlists`, qui affiche les listes non terminées sur lesquelles il travaille.

text input levels T_EX peut travailler sur des documents sources qui s'appellent en cascade (par exemple avec `\include`, `\input` ou `\usepackage`). Sur la machine d'un des auteurs, il faut emboîter 1500 fichiers pour atteindre la limite.

B.2 Messages d'avertissement et d'information

Tandis que les messages d'erreur arrêtent L^AT_EX et attendent une réponse de l'utilisateur, les messages d'avertissement sont juste affichés sur le terminal et dans le fichier de trace, et le traitement se poursuit. S'il le peut, L^AT_EX montre également le numéro de la ligne du source qui a déclenché l'avertissement. Les avertissements du noyau L^AT_EX sont précédés d'un «LaTeX Warning: » ou d'un «LaTeX Font Warning: ». Autrement, les avertissements indiquent l'extension ou la classe dont ils sont issus avec «Package *<nom>* Warning: » ou «Class *<nom>* Warning: ». Les avertissements de T_EX, comme «Overfull... », ne sont pas précédés d'une chaîne standard.

Outre les avertissements, L^AT_EX écrit des messages d'information dans le fichier de trace sans les afficher à l'écran. Pour bien distinguer les messages d'avertissement et d'information, nous afficherons les [avertissements](#) en bleu dans la liste alphabétique suivante.

Calculating math sizes for size *<taille-texte>*

L^AT_EX doit deviner les tailles correctes de fontes pour les exposants et les indices, car il ne trouve pas cette information pour la *<taille-texte>* dans ses tables internes. Ce message est généralement suivi de plusieurs avertissements de correction de tailles de fontes, car la supposition initiale de L^AT_EX est rarement la bonne. Cette situation survient lorsqu'on sélectionne une taille inhabituelle avec la commande `\fontsize`. Voir section 7.10.7 si les formules mathématiques semblent étranges.

Checking defaults for *<code>/<forme-fonte>*

Ce message est écrit dans le fichier de trace au moment du `\begin{document}`, pendant que L^AT_EX vérifie que les substitutions par défaut pour le codage *<code>* sont cohérentes. Il est suivi d'un «...okay » ou d'un message d'erreur lorsque L^AT_EX ne connaît pas le groupe de *<forme-fonte>* spécifié avec `\DeclareFontEncoding`.

Citation '*<clé>*' on page *<numéro>* undefined

La *<clé>*, spécifiée en tant qu'argument de `\cite` ou de `\nocite`, n'est pas définie par une commande `\bibitem`, ou nécessite une autre compilation L^AT_EX (éventuellement B_BL_AT_EX également) pour que L^AT_EX la reconnaisse. Dans le dernier cas, on obtient l'avertissement supplémentaire «Label(s) may have changed... », comme on le verra à la page 941. Le *<numéro>* de page est absent de l'avertissement si celui-ci est émis par `\nocite`.

¹Les auteurs n'arrivent même pas à imaginer une définition à problème qui n'atteindrait pas d'abord une autre limite.

Command $\langle nom \rangle$ invalid in math mode

Il s'agit d'un avertissement ou d'une erreur qui indique qu'on a utilisé une commande en mode mathématique, alors qu'elle ne doit être spécifiée qu'en mode texte. Il s'agit d'un avertissement lorsque la construction en question est périmée, mais toujours valide.

Document Class : $\langle nom \rangle$ $\langle date \rangle$ $\langle info-supplémentaire \rangle$

Cette ligne est produite par une commande `\ProvidesClass` utilisée au niveau du code de la classe du document. Bien que ce ne soit pas un avertissement, elle apparaît à la fois sur le terminal et dans le fichier de trace. Si un document produit différents résultats sur différentes installations, on doit comparer les messages « Document Class : », « File : » et « Package : » pour identifier les différences de versions.

Empty 'thebibliography' environment

Cet avertissement est dû à un environnement `thebibliography` sans commande `\bibitem`. Il indique souvent un problème avec une compilation \LaTeX , par exemple, lorsque le programme \LaTeX n'a pas été capable de résoudre une citation isolée.

Encoding $\langle nom \rangle$ has changed to $\langle autre-nom \rangle$ for ...

Cet avertissement est émis lorsqu'une déclaration d'une fonte de symboles utilise différentes méthodes de codage dans différentes versions mathématiques. Il peut indiquer que des commandes `\DeclareMathSymbol` pour cette fonte de symboles ne sont pas valides dans toutes les versions mathématiques.

 $\langle \end \rangle$ occurred $\langle lorsque \rangle$ \TeX

On reçoit cet avertissement en toute fin de compilation à chaque fois que \TeX trouve que le `\end{document}` ou la commande `\stop` est prématuré. Cet avertissement est malheureusement ambigu, puisqu'il fait référence à la primitive \TeX `\end` qui est redéfinie par \LaTeX pour devenir la balise de fin d'environnement. Le $\langle lorsque \rangle$ peut être :

inside a group at level $\langle numéro \rangle$ \TeX

Dans ce cas, la compilation \LaTeX se termine alors qu'il y avait encore des groupes ouverts. Ces groupes peuvent être des accolades explicites non refermées (par exemple, `{\itshape. .}`), l'utilisation de `\bgroup` ou de `\begingroup` dans une macro sans les fermetures correspondantes, ou des environnements non terminés dans le source. Normalement, ce dernier cas déclenche d'abord une erreur \LaTeX plus précise (c'est-à-dire « `{\langle env \rangle}` on . . . »), sauf si la compilation se termine par `\stop`. Dans ce cas, la vérification de mauvaise correspondance de fin d'environnement n'est pas effectuée.

when $\langle condition \rangle$ on line $\langle numéro \rangle$ was incomplete) \TeX

\LaTeX a achevé la compilation avec une condition \TeX de bas niveau non terminée. Dans les documents \LaTeX qui n'utilisent que les commandes standard, ce problème ne devrait jamais arriver, à moins de terminer le document au niveau d'un fichier chargé par `\include`. Dans les autres cas, il indique sans doute un

bogue dans une extension. On peut essayer d'identifier l'origine de la condition (en regardant le *numéro* de ligne) pour voir quelle commande a été utilisée. On notera que le *numéro* de ligne peut se référer à un autre fichier que le source (malheureusement, TeX ne donne pas le nom de fichier). Dans des situations très difficiles, on peut essayer les options de traçage de eTeX pour repérer le problème : en donnant la valeur 1 à `\tracingifs`, on obtient une trace détaillée sur la façon dont des conditions emboîtées sont exécutées.

External font *nom* loaded for size *taille*

TeX ignore une requête de chargement d'une forme de fonte à une certaine *taille* et charge la fonte *nom* à la place. Ce message est produit par la fonction de taille `fixed`.

Faking *commande* for font family *nom* in TS1 encoding `textcomp`

Le glyphe correspondant à la *commande* n'est pas disponible dans le codage TS1 de la famille de fontes en cours. TeX fournit un caractère qu'il aura « construit » d'une certaine façon. C'est, par exemple, le cas du glyphe `\texteuro` (€) lorsqu'il n'est pas disponible. La section 7.8.7 a décrit les différentes possibilités d'obtenir un véritable signe euro.

File '*nom*' already exists on the system.

Not generating it from this source

Cet avertissement est produit par un environnement `filecontents` lorsque le fichier *nom* existe déjà dans le chemin de recherche de TeX. Si l'on veut quand même créer le fichier, il faut supprimer (ou renommer) la version trouvée par TeX, ou extraire le fichier manuellement avec un éditeur de texte.

File : *nom* *date* *info-supplémentaire*

Cette ligne est produite par la commande `\ProvidesFile`, qui est utilisée pour identifier un fichier, ainsi que sa dernière date de modification. Par convention, l'*info-supplémentaire* commence par un numéro de version, même si ce n'est pas une obligation stricte. Bien que de même importance que celle de `\ProvidesClass`, cette information est seulement écrite au niveau du fichier de trace, afin de ne pas surcharger le terminal en messages. Si un document produit différents résultats sur différentes installations, on doit comparer les messages « Document Class: », « File: » et « Package: » pour identifier les différences de versions.

File : *codage**famille*.fd *date* *info-supplémentaire*

Il s'agit d'un cas spécial du message d'information précédent, qui indique que le fichier de définition de fontes pour une certaine combinaison de *codage* (normalement affiché en minuscule) et de *famille* a été chargé. Ces fichiers contiennent des déclarations de groupes de formes et sont décrits à la section 7.10.6.

Float too large for page by *hauteur*

Un flottant est trop grand de *hauteur* pour pouvoir loger dans le `\textheight` en cours. Il sera composé seul sur une page (si c'est permis), en débordant éventuellement au niveau de la marge de pied. Si le flottant n'a pas la permission d'aller sur une page flottante, il empêchera de placer tous les autres flottants de sa classe.

Font shape $\langle forme \rangle$ in size $\langle taille \rangle$ not available

ℒ_{TEX} émet ce message lorsqu'il essaie de sélectionner une fonte pour laquelle la combinaison d'attributs n'est pas disponible et lorsqu'une substitution est définie dans les tables internes. Selon le contenu de ces tables, l'un des messages suivants s'affichera :

external font $\langle nom \rangle$ used

ℒ_{TEX} a sélectionné la fonte externe $\langle nom \rangle$ dans cette situation particulière et il ne sait pas à quel groupe de formes elle appartient. Ce message est produit par la fonction de taille subf.

size $\langle taille \rangle$ substituted

ℒ_{TEX} a sélectionné la forme correcte, mais puisque la taille demandée n'est pas accessible, ℒ_{TEX} a choisi la $\langle taille \rangle$ la plus proche. Cette action est automatique si aucune des tailles individuelles ou des intervalles de tailles de la déclaration de $\langle forme \rangle$ de fontes ne correspondent.

shape $\langle forme \rangle$ tried

ℒ_{TEX} a sélectionné un groupe de $\langle forme \rangle$ de fontes différent, car celui demandé n'est pas disponible pour la $\langle taille \rangle$ demandée. Ce message est produit par la fonction de taille sub.

Font shape $\langle forme \rangle$ undefined. Using ' $\langle autre-forme \rangle$ ' instead

Cet avertissement indique qu'une combinaison d'attributs de fontes n'a pas de définition de formes de fontes correspondante. Par exemple, demander `\fontseries{b}\ttfamily` déclenche normalement cet avertissement, puisque les fontes Computer Modern n'ont ni typewriter gras, ni typewriter gras étendu. Cela dit, lorsque cette dernière combinaison est demandée, on ne reçoit pas d'avertissement mais seulement un message d'information au niveau du fichier de trace, car pour `\textbf{\texttt{. .}}`, les fichiers .fd contiennent une règle explicite de substitution.

Si ℒ_{TEX} rencontre un symbole particulier qui ne peut pas être composé avec la forme de fontes demandée, l'avertissement ci-dessus est suivi d'un « for symbol $\langle nom \rangle$ ».

Font shape $\langle forme \rangle$ will be scaled to size $\langle taille \rangle$

ℒ_{TEX} a chargé la fonte demandée en effectuant un changement d'échelle pour la $\langle taille \rangle$ désirée. Pour imprimer un document contenant des fontes mises à l'échelle, le pilote d'impression doit avoir ces fontes à la taille correcte ou doit être capable de les agrandir ou de les diminuer automatiquement.

Foreign command $\langle commande \rangle$; amsmath
 \frac or \genfrac should be used instead

Bien que l'utilisation de la $\langle commande \rangle$ spécifiée ne soit pas considérée comme une erreur, il est fortement déconseillé d'utiliser ces anciennes formes pour les fractions et les fractions généralisées sous ℒ_{TEX}. Il est préférable d'utiliser les commandes \frac et \genfrac de l'extension amsmath.

Form feed has been converted to Blank Line

L'environnement `filecontents` a détecté un caractère « fin de ligne » (\char"000A) dans le source et le traduit par une ligne vide (commande `\par` si c'est \TeX qui l'interprète) au niveau du fichier externe. Comme `filecontents` a été conçu pour produire des données de texte, cette commande ne peut pas être utilisée pour gérer des fichiers binaires.

`'h' float specifier changed to 'ht'` ou
`'!h' float specifier changed to '!ht'`

On a spécifié un placement de flottant avec `h` ou `!h` sans donner d'autres options. \TeX demande une autre possibilité au cas où le « here » (ici) se révélerait impossible à satisfaire en raison d'un manque de place sur la page en cours. Si l'on veut réellement empêcher les flottants de flotter, on peut utiliser l'extension `float`, décrite à la section 6.3.1.

Ignoring text '*texte*' after `\end{env}`

Cet avertissement est émis par `filecontents` ou `filecontents*` lorsqu'il y a du texte à la suite de la balise `\end`.

Label '*clé*' multiply defined

Le document comporte plusieurs commandes `\label` qui utilisent la même *clé*. Les références à cette *clé* se feront toujours sur la dernière définie. Il faut vérifier que toutes les clés sont différentes.

Label(s) may have changed. Rerun to get cross-references right

\TeX a détecté des modifications sur les définitions d'étiquettes par rapport à la compilation précédente et demande (au moins) une compilation supplémentaire pour résoudre correctement les références croisées.

En théorie, il est possible, bien que peu probable, que ce message persiste quel que soit le nombre de compilations effectuées¹. Si c'est le cas, il faut comparer les fichiers `.aux` de deux compilations successives pour trouver l'étiquette qui alterne entre deux pages, puis résoudre le problème à la main.

Loose `\hbox` (badness *nombre*) *quelque-part* \TeX

\TeX a produit une boîte horizontale de médiocrité 13 ou plus (ce qui correspond à l'utilisation de 50% ou plus de l'étirement disponible). Cet avertissement peut être ignoré sans problème, à moins d'être perfectionniste. En fait, il ne sera pas produit si l'on ne modifie pas la valeur par défaut de `\hbadness`. Voir le message « `Underfull \hbox...` » à la page 946 pour plus de détails.

Loose `\vbox` (badness *nombre*) *quelque-part* \TeX

\TeX a produit une boîte verticale de médiocrité 13 ou plus (ce qui correspond à l'utilisation de 50% ou plus de l'étirement disponible). L'avertissement est produit seulement si l'on a affecté une valeur inférieure à 100 à `\vbadness`. Voir le message « `Underfull \vbox...` » à la page 947 pour plus de détails.

¹Par exemple, lorsqu'une commande `\label` est proche de la frontière entre les deux pages « iii » et « iv », l'utilisation d'une commande `\pageref` avant `\label` peut donner une situation où `\label` sera placé en page « iv » si la référence est « iii » et vice versa.

Making \langle caractère \rangle an active character LaTeX

Cette information est écrite dans le fichier de trace pour chaque caractère qui devient un raccourci. Lorsqu'un document montre des résultats non prévus, cette information peut être utile si les problèmes sont dus à l'utilisation par inadvertance de caractères de raccourci.

Marginpar on page \langle numéro \rangle moved

Cet avertissement indique qu'une commande `\marginpar` n'est pas alignée avec la ligne de texte dont elle dépend, en raison d'un `\marginpar` précédent qui prend la place.

Missing character : There is no \langle caractère \rangle in font \langle nom \rangle ! LaTeX

Bien que ce message indique généralement un sérieux problème, il ne s'agit malheureusement que d'un message d'avertissement et il n'est donc écrit que dans le fichier de trace (à moins que `\tracingonline` soit strictement positif). Il signifie que, quelque part dans le source, une demande de symbole \langle caractère \rangle a été faite pour laquelle la fonte en cours (de nom externe \langle nom \rangle) n'a pas de glyphe dans la position correspondante. Le \langle caractère \rangle affiché peut varier selon la distribution \TeX utilisée¹. Par exemple, l'utilisation de la commande `\symbol` peut déclencher cet avertissement, puisque cette commande permet d'accéder à n'importe quelle position de n'importe quelle fonte. En revanche, les *font-encoding-specific commands* standard, présentées à la section 7.11.4 page 463, ne devraient jamais produire cet avertissement.

Rechercher
ce message dans le
fichier log!

No `\author given`

On a utilisé la commande `\maketitle` sans spécifier d'auteur au préalable. Contrairement à l'absence de `\title`, celle-ci ne produit qu'un avertissement.

No auxiliary output files

Cet avertissement s'affiche lorsqu'on utilise la déclaration `\nofiles` au niveau du préambule du document.

No characters defined by input encoding change to \langle nom \rangle

Le fichier de codage d'entrée \langle nom \rangle .def ne semble contenir aucune déclaration de codage d'entrée. Pour le codage `ascii`, il s'agit d'un comportement normal, mais pour les autres codages, cet avertissement indique un problème.

No file \langle nom \rangle

\TeX affiche cette information à chaque fois qu'il essaie de lire un fichier auxiliaire (`.aux` ou `.toc` par exemple), mais qu'il ne peut pas le trouver. Ce n'est pas considéré comme une erreur, puisque ces fichiers ne sont créés qu'après la première compilation. Cependant, la même routine est malheureusement utilisée par la commande `\include`, et donc un fichier à inclure n'existant pas déclenche le même avertissement, alors qu'il s'agit là d'une erreur.

¹Quelquefois, on voit quelque chose comme `^^G`, d'autres fois, des caractères réels sont affichés. Malheureusement, il n'y a aucune garantie qu'ils correspondent au source : au niveau du fichier de trace, on a une certaine traduction, qui dépend du système d'exploitation.

No hyphenation patterns were loaded for the language '*⟨langue⟩*' babel

Tous les fichiers de définition de langues vérifient si les motifs de césures sélectionnés sont chargés ou non dans le format \TeX . Si ce n'est pas le cas, ce message s'affiche et des motifs de césures par défaut sont utilisés. Les motifs par défaut sont ceux chargés dans le registre de motifs 0 (habituellement l'anglais américain).

No input encoding specified for *⟨langue⟩* language babel

Ce message peut apparaître lorsqu'on n'a demandé aucun codage d'entrée spécifique pour le document, alors qu'un des supports de langues nécessite l'alphabet cyrillique pour la composition. Plusieurs codages d'entrée existent pour ces langues. Par conséquent, la définition de langues insiste en disant que celui qui est utilisé *doit* être explicitement indiqué.

No positions in optional float specifier. Default added ...

On a utilisé un environnement flottant (par exemple, `figure` ou `table`) avec son argument optionnel de placement qui ne contient aucune information exploitable. Dans ces conditions, \TeX utilise ses règles de placement par défaut.

Oldstyle digits unavailable for family *⟨nom⟩* textcomp

On a utilisé `\oldstylenums` avec une famille de fontes qui ne contient aucun chiffre elzévirien. \TeX prend une mesure d'urgence qui consiste à produire des chiffres modernes (pris dans une autre famille de fontes) à la place. Voir section 7.5.4 pour plus de détails.

Optional argument of `\twocolumn` too tall on page *⟨numéro⟩*

Le matériel dans l'argument optionnel de `\twocolumn` est tellement long qu'il ne reste plus que trois lignes ou moins dans la page. \TeX ne commencera pas le mode double-colonnes sur cette page, mais débutera une nouvelle page.

`\oval`, `\circle`, or `\line size` unavailable

La taille demandée pour les commandes indiquées n'est pas disponible. \TeX choisit la taille la plus proche. Pour éviter ce problème, on peut se reporter, par exemple, à la section 10.4.3.

Overfull `\hbox` (*⟨numéro⟩*pt too wide) *⟨quelque-part⟩* TeX

\TeX a été obligé de construire une boîte horizontale (par exemple, une ligne de paragraphe ou un `\makebox`) d'une certaine longueur et il n'a pas été capable de faire tenir le résultat dans la largeur donnée, même après avoir compressé au maximum tous les ressorts. En conséquence, la boîte déborde sur sa droite. Dans la plupart des cas, c'est assez visible, même si le débordement est faible. On doit corriger ce problème manuellement, puisque \TeX est incapable de le résoudre lui-même (les sections 3.1.11 et B.3.3 donnent quelques conseils). Pour une liste et une explication des causes possibles (c'est-à-dire le *⟨quelque-part⟩*), voir l'avertissement «`Underfull \hbox...`» à la page 946.

Overfull `\vbox` (*⟨nombre⟩*pt too wide) *⟨quelque-part⟩* TeX

On a demandé à \TeX de construire une boîte verticale d'une taille fixée (en utilisant, par exemple, `\parbox` ou `minipage` avec un deuxième argument optionnel ;

voir annexe A.2.2 page 880), et le matériel ne peut y tenir. Le matériel en trop débordera en bas de la boîte. Les problèmes que cela peut poser dépendent des circonstances. Pour une liste et une explication des causes possibles (c'est-à-dire le *«quelque-part»*), voir l'avertissement «Underfull \vbox...» à la page 947.

Overwriting encoding scheme *«quelque-chose»* defaults

Cet avertissement est émis par `\DeclareFontEncodingDefaults` lorsque cette commande remplace des codages par défaut pour le texte ou les mathématiques qui ont déjà été déclarés.

Overwriting *«quelque-chose»* in version *«nom»* ...

Une déclaration, telle que `\SetSymbolFont` ou `\DeclareMathAlphabet`, a modifié la forme de fontes en affectant la valeur *«quelque-chose»* (une fonte de symboles ou un alphabet mathématique) à la version mathématique *«nom»*.

Package : *«nom» «date» «info-supplémentaire»*

Ce message d'information est produit par la commande `\ProvidesPackage`, qui est utilisée pour identifier une extension en indiquant également sa dernière date de modification. Par convention, l'*«info-supplémentaire»* commence par le numéro de version, mais il ne s'agit pas d'une obligation. Bien que de la même importance que `\ProvidesClass`, cette information n'est écrite que dans le fichier de trace afin d'éviter de remplir le terminal avec des messages. Si un même document produit différents messages sur différentes installations, on doit comparer les messages «Document Class:», «File:» et «Package:» pour identifier les différences de versions.

Redeclaring font encoding *«nom»*

Cet avertissement est émis lorsqu'on utilise `\DeclareFontEncoding` pour définir un codage déjà existant (qui peut potentiellement modifier les valeurs par défaut).

Redeclaring math accent *«nom»*

Cet avertissement est émis lorsque `\DeclareMathAccent` est utilisé pour un accent mathématique déjà défini. Si la commande à déclarer est connue mais n'est pas un accent, on obtient un message d'erreur.

Redeclaring math alphabet *«nom»*

On a appelé `\DeclareMathAlphabet` ou `\DeclareSymbolFontAlphabet` afin de définir l'alphabet *«nom»*, alors que ce dernier était déjà défini en tant qu'identificateur d'alphabet mathématique. La nouvelle déclaration remplace tous les anciens réglages pour *«nom»*.

Redeclaring math symbol *«nom»*

La commande *«nom»* a déjà été déclarée en tant que symbole mathématique, et cette nouvelle déclaration remplace l'ancienne définition.

Redeclaring math version *«nom»*

On a utilisé `\DeclareMathVersion` pour une version mathématique déjà déclarée. Cette déclaration va remplacer tous les anciens réglages de cette version par les nouvelles valeurs par défaut.

Redeclaring symbol font *<nom>*

On a utilisé une commande `\DeclareSymbolFont` pour une fonte de symboles qui était déjà déclarée. La nouvelle déclaration remplace la fonte de symboles dans toutes les versions mathématiques connues.

Reference '*<clé>*' on page *<numéro>* undefined

Une référence créée avec une commande `\ref`, `\pageref`, ou l'une des commandes de références croisées étudiées au chapitre 2, utilise une *<clé>* pour laquelle \TeX ne trouve pas de commande `\label` correspondante. Si ce `\label` est quelque part dans le document, il suffit simplement de faire une compilation supplémentaire pour que \TeX puisse le reconnaître. Cette situation est précisée par l'avertissement supplémentaire «Label(s) may have changed...», vu à la page 941.

Size substitutions with differences up to *<taille>* have occurred

Ce message apparaît à la fin d'une compilation, lorsque \TeX sélectionne au moins une taille de fontes sensiblement différente de celle demandée et qui n'existe pas. La *<taille>* est la différence maximale qui a été nécessaire.

Some font shapes were not available, defaults substituted

Ce message apparaît à la fin d'une compilation, lorsque \TeX a été obligé d'utiliser des substitutions automatiques pour certaines formes de fontes.

Tab has been converted to Blank Space

L'environnement `filecontents` a détecté un caractère «*Tab*» (\char"0009) dans le source et l'écrit sous la forme d'un caractère espace dans le fichier externe.

Text page *<numéro>* contains only floats

Un ou plusieurs flottants, traités en tant que flottants de type « en haut de page » ou « en bas de page », occupent une place tellement importante qu'il reste très peu d'espace sur la page en cours (une ou deux lignes) pour le texte normal. \TeX décide alors de ne placer que les flottants sur la page en question (même si certains ou tous les flottants ne permettent pas explicitement ce type de placement). Ce message n'apparaît que lorsque les paramètres de placement des flottants sont beaucoup plus exigeants que leurs valeurs par défaut. On se reportera au chapitre 6 pour plus de détails.

There were multiply-defined labels

Cet avertissement apparaît à la fin de la compilation, lorsque \TeX détecte au moins deux commandes `\label` ou `\bibitem` qui ont la même clé. Pour localiser l'erreur, il faut examiner le fichier de trace, puis s'assurer que toutes les clés sont différentes.

There were undefined references

Cet avertissement apparaît à la fin de la compilation, lorsque \TeX détecte des références à des clés inconnues et voit qu'une nouvelle compilation ne résoudra pas le problème. Il faut examiner, dans le fichier de trace, toutes les occurrences de «Reference *<clé>* undefined» et «Citation *<clé>* undefined» et les corriger (il peut s'agir d'une faute de frappe sur la clé ou de l'absence d'une commande `\label` ou `\bibitem`). Dans le cas d'une citation *<clé>* absente, il suffit de relancer `\BIBTeX`, puis \TeX .

Tight \hbox (badness <nombre>) <quelque-part> TeX

TeX a produit une boîte horizontale et doit comprimer les espaces intérieurs au-delà d'une certaine limite. On voit ce message uniquement lorsque `\hbadness` a une valeur strictement inférieure à 100. Voir le message «`Underfull \hbox...`» ci-dessous pour plus de détails.

Tight \vbox (badness <nombre>) <quelque-part> TeX

TeX a produit une boîte verticale et doit comprimer les espaces intérieurs au-delà d'une certaine limite. On voit ce message uniquement lorsque `\vbadness` a une valeur strictement inférieure à 100. Voir le message «`Underfull \vbox...`» page suivante pour plus de détails.

Try loading font information for <code>+<famille>

Ce message est envoyé dans le fichier de trace à chaque fois que \LaTeX essaie de charger un fichier `.fd` pour la combinaison codage-famille `<code>/<famille>` d'une fonte.

Unable to redefine math accent <accent> amsmath

Cet avertissement est rare. Il est émis lorsqu'on charge l'extension `amsmath` avec des fontes mathématiques non standard.

Underfull \hbox (badness <nombre>) <quelque-part> TeX

On a obligé TeX à construire une boîte horizontale (par exemple, la ligne d'un paragraphe ou un `\makebox`) d'une certaine largeur et l'espace à l'intérieur de cette boîte a dû s'étirer plus que ce qui était admis (c'est-à-dire un étirement de plus de 100% de la somme des parties plus des ressorts). En interne, cette situation est exprimée sous la forme d'une médiocrité supérieure à 100. Par exemple, une valeur de 800 indique qu'il a fallu une augmentation de 200%¹.

Le fait de savoir si cette boîte incomplète constitue effectivement un réel problème est une affaire de jugement visuel au niveau de la sortie. Si la médiocrité est égale à 10000, la boîte peut être arbitrairement mauvaise. Comme la valeur de TeX pour l'infini est assez basse, il aura tendance à favoriser une boîte vraiment horrible par rapport à plusieurs boîtes successives mauvaises mais acceptables. Dans ce cas, l'utilisation de `\emergencystretch` peut être utile (voir la section 3.1.11).

La valeur limite de la médiocrité pour laquelle ces avertissements s'afficheront est indiquée par le paramètre entier `\hbadness`. La valeur par défaut sous \LaTeX est de 1000. Les avertissements n'apparaissent donc que pour des boîtes réellement mauvaises. Lorsqu'on veut produire un document volumineux, on peut essayer une valeur plus restrictive, comme `\hbadness=10`, pour voir combien de lignes seront jugées imparfaites par TeX.

On notera que l'avertissement parle toujours de `\hbox`, sans tenir compte de la construction effectivement utilisée dans le source, puisque il est directement produit par TeX. L'emplacement où le problème survient est indiqué par le *<quelque-part>*. Il peut s'agir d'une des quatre possibilités suivantes :

¹La formule exacte est $\min(100r^3, 10000)$, où r est le rapport entre l'étirement utilisé et l'étirement permis, sauf si un ressort d'étirement infini existe (indiqué, par exemple, avec une commande `\hfill`), auquel cas la médiocrité est nulle.

`detected at line <numéro>` TeX

Le problème provient d'une construction de boîte explicite (`\makebox` avec un argument de longueur explicite, par exemple, ou tout autre construction \LaTeX construisant des boîtes), qui se termine à la ligne *<numéro>* dans le source.

`has occurred while \output is active` TeX

\TeX était en train de construire une page et a rencontré un problème en plaçant l'en-tête, le pied de page ou d'autres éléments de ce type. Comme il s'agit d'une opération asynchrone, aucun numéro de ligne n'est donné. Il faut regarder la page produite la plus proche de celle où se situe l'avertissement pour savoir s'il est nécessaire de procéder à une correction manuelle.

`in alignment at lines <numéros>` TeX

La boîte est une partie d'un environnement `tabular` ou d'un alignement mathématique. Les *<numéros>* de ligne donnent la position de la structure complète dans le source, puisqu'au moment où \TeX rencontre le problème, il n'a plus aucun moyen d'examiner le source en détail.

`in paragraph at lines <numéros>` TeX

La boîte incomplète est due à de mauvaises espaces dans une ligne d'un paragraphe (indiqué par les *<numéros>* de ligne). L'affichage symbolique de la ligne en question est utile pour repérer précisément le problème.

`Underfull \vbox (badness <nombre>) <quelque-part>` TeX

On a obligé \TeX à construire une boîte verticale (par exemple, une `\parbox` ou une `minipage`) d'une certaine hauteur et l'espace vertical à l'intérieur de cette boîte a dû s'étirer plus que ce qui était admis. On se reportera à la description de l'avertissement «`Underfull \hbox...`» pour plus de détails. On peut supprimer tous les avertissements pour les valeurs de médiocrité au-dessous d'une certaine limite avec l'affectation `\badness=<valeur>`. \TeX n'émettra alors des avertissements que pour les boîtes qui ont une médiocrité supérieure à *<valeur>* (la valeur par défaut est de 1000). Le *<quelque-part>* indique l'origine du problème et peut être l'un des cas suivants :

`detected at line <numéro>` TeX

La boîte a été explicitement construite (le *<numéro>* de ligne indique la fin de la construction) et il n'y a pas assez d'étirement de ressort disponible. Par exemple,

```
\parbox[c][2in][s]{4cm}{test test}
```

produit cet avertissement, car la boîte doit faire deux pouces de haut et son contenu doit remplir cette hauteur (argument *s*), mais il n'y a rien qui puisse être étiré, par exemple, quelque chose comme `\par\vfill` entre les deux mots. Voir annexe A.2.2 pour plus d'informations sur les boîtes verticales.

`has occurred while \output is active` TeX

Dans le cas le plus fréquent, les espaces sur la page en cours ont dû être étirés au-delà des limites acceptables aux yeux de \TeX . Que cela représente effectivement ou non un problème visuel dépend de nombreux facteurs, tels que le

type des espaces de la page. Par exemple, un grand blanc au niveau d'un titre est habituellement moins ennuyeux qu'une liste avec des lignes très espacées. Le meilleur conseil que l'on puisse donner est de vérifier ces pages soi-même. Les commandes `\enlargethispage` ou `\pagebreak` sont souvent utiles.

Si le problème apparaît étonnamment souvent, les paramètres d'espace-ment des listes, des paragraphes et des titres devraient être vérifiés pour savoir s'ils ne sont pas trop rigides (voir chapitres 2 à 4). On pourra également vérifier si le `\textheight` correspond à un nombre entier de lignes de texte. Voir la présentation page 201.

`in alignment at lines <numéro-lignes>` TeX

Cet avertissement ne devrait pas arriver en \TeX standard, mais il peut survenir avec des applications spéciales. Dans ce cas, il faut utiliser `<numéro-lignes>` pour identifier les lignes du source à problème dans le document.

`Unused global option(s) : [liste-options]`

Quelques options spécifiées au niveau du `\documentclass` n'ont été utilisées ni par la classe, ni par les extensions indiquées dans le préambule. La raison la plus probable est que les noms de ces options ont été mal saisis. On notera également que certaines extensions ne réagissent pas aux options globales, mais seulement à celles explicitement spécifiées lors de leur chargement (voir annexe A.4).

`Writing file '<nom>'`

Ce message d'information est produit par `filecontents` et `filecontents*` lorsque ces environnements écrivent leur corps dans le fichier externe `<nom>`.

`Writing text '<texte>' before \end{<env>} as last line of <fichier>`

Les environnements `filecontents` et `filecontents*` émettent cet avertissement lorsqu'ils détectent du texte immédiatement avant la balise `\end`.

`You have more than once selected the attribute '<attribut>'` babel
`for language <langue>`

Ce message s'affiche lorsque le même `<attribut>` est utilisé plusieurs fois dans le deuxième argument d'une commande `\languageattribute`. Seule la première occurrence déclenche l'activation de l'`<attribut>`.

`You have requested <package/class> '<nom>',`
`but the <package/class> provides '<autre-nom>'`

On a demandé le chargement de `<nom>` par `\usepackage` ou `\RequirePackage` (dans le cas d'une extension), ou par `\documentclass` ou `\LoadClass` (dans le cas d'une classe), mais l'extension ou la classe indique un `<autre-nom>` que celui demandé. S'il ne s'agit pas d'une erreur de frappe du concepteur de l'extension ou de la classe, cela indique que l'installation comporte une variante de l'extension ou de la classe, qui se comporte sans doute différemment de l'originale. Par conséquent, le document pourra être composé d'une manière différente lorsqu'il sera traité sur une autre installation. Même si le comportement est correct, il s'agit d'un point qui nécessite d'être éclairci en examinant l'extension ou la classe en question.

Spécifier un nom de chemin relatif ou absolu pour effet de bord de déclencher cet avertissement.

You have requested release '*<date>*' of LaTeX,
but only release '*<ancienne-date>*' is available

Une commande `\NeedsTeXFormat` a demandé une version L^AT_EX qui date d'au moins *<date>*, mais la date du format L^AT_EX présent est *<ancienne-date>*. Habituellement, une telle demande est faite pour s'assurer que certaines fonctionnalités du format L^AT_EX sont bien disponibles. Il est donc probable que le document produira des erreurs ultérieures ou un résultat bizarre. Il convient de mettre à jour le format L^AT_EX de la distribution.

You have requested, on line *<numéro>*, version '*<date>*' of *<nom>*,
but only version '*<ancienne-date>*' is available

On a demandé une classe ou une extension dont la date est au moins *<date>*, mais la version sur l'installation date de *<ancienne-date>*. Il convient de mettre à jour la classe ou l'extension en question.

B.3 Commandes T_EX et L^AT_EX de trace

Dans cette section, nous présenterons les outils et les techniques qui permettent de tracer et d'afficher des informations sur l'état du système – par exemple, trouver pourquoi une structure produit des espaces parasites sur la page ou pourquoi ses propres définitions de commandes ne fonctionnent pas correctement.

B.3.1 Afficher les définitions de commandes et les valeurs de registres

Dans de nombreuses situations, il est utile d'obtenir des informations sur les valeurs internes en cours de L^AT_EX : les définitions précises de commandes, les valeurs de registres, etc. Par exemple, si l'utilisation de `\newcommand` indique que la commande qui doit être définie existe déjà, on peut vouloir connaître sa définition en cours pour être certain que l'on ne cherche pas à redéfinir une commande importante.

Dans ce but, T_EX offre la commande `\show` qui affiche la définition de l'unité lexicale qui la suit, et qui affiche un point d'interrogation en attendant une intervention de l'utilisateur. Par exemple, après avoir défini `\xvec` comme dans l'exemple A-1-4 page 859, on peut afficher sa définition comme suit :

*Afficher une
définition de
commande*

```
\newcommand\xvec[1]{\ensuremath{x_1,\ldots,x_{#1}}}  
\show\xvec
```

Cela produira la sortie suivante sur le terminal et dans le fichier de trace :

```
> \xvec=\long macro:  
#1->\ensuremath {x_1,\ldots ,x_{#1}}.  
1.6 \show\xvec  
  
?
```

La première ligne, qui débute par `>`, montre l'unité lexicale en train de s'afficher (`\xvec`) et donne son type (`\long macro`), qui indique que `\xvec` est une macro qui accepte des commandes `\par` dans ses arguments. Autrement dit, cette macro a été définie par `\newcommand` et non par `\newcommand*`. La dixième ligne montre la structure des arguments de cette commande (jusqu'à `->`), ce qui indique que la commande a un argument (`#1`). On notera que l'argument de la déclaration `\newcommand`, qui était indiqué par `[1]`, s'affiche maintenant de façon différente. Le reste de la ligne – éventuellement de plusieurs lignes, si nécessaire – montre la partie définition. Le code se termine par un point qui ne fait pas partie de la définition mais qui aide à identifier d'éventuels espaces en fin de définition. On peut noter que le code a été normalisé. Ainsi, après une commande qui avale les espaces qui suivent, on voit un espace, que la définition d'origine ait ou non indiqué cet espace (c'est le cas dans l'exemple avec l'espace qui suit la commande `\ldots` et qui n'existait pas dans la définition de `\xvec`).

La ligne du source (avec son numéro de ligne dans le fichier source) s'affiche à la suite de la définition. \TeX s'arrête en affichant un point d'interrogation. Pour continuer, il suffit d'appuyer sur la touche *(Entrée)*. On peut également appuyer sur la touche `h` pour voir quelles sont les autres possibilités.

Pendant, toutes les commandes ne produisent pas de sortie aussi facilement compréhensible. Par exemple, si l'on affiche une commande qui a été définie avec un argument optionnel, comme la commande `\lvec` de l'exemple A-1-5 page 859,

```
\newcommand\lvec[2][n]
  {\ensuremath{#2_1+\cdots + #2_{#1}}}
\show\lvec
```

on obtient le résultat suivant :

```
> \lvec=macro:
->\@protected@testopt \lvec \lvec {n}.
```

Visiblement, la commande `\lvec` n'a pas d'arguments (ils sont lus plus tard dans le traitement). Il y a autre chose d'étrange dans cette information : qu'est-ce donc que ce `\lvec` ? Est-ce la commande `\l` suivie des lettres `lvec` ou est-ce une commande étrange `\lvec` ayant deux contre-obliques dans son nom ? Il s'agit effectivement de cette dernière possibilité, mais il est impossible de le déterminer en se basant uniquement sur l'affichage proposé par la commande `\show`. Ces noms de commandes bizarres, qui ne peuvent pas être produits facilement par l'utilisateur, sont parfois utilisés de façon interne par \TeX pour produire de nouveaux noms de commandes à partir de noms existants, en utilisant `\csname` et `\endcsname`, ainsi que d'autres mécanismes de bas niveau de \TeX .

*Afficher des
commandes internes
avec des noms
bizarres*

Que doit-on faire si l'on veut voir la définition de `\lvec` ? Il doit être clair qu'écrire `\show` devant cette commande ne fonctionnera pas puisque, en situation normale, \TeX voit `\l` et pense qu'il s'agit de la commande à montrer. Pour cette raison, on doit absolument utiliser les mêmes mécanismes de bas niveau qui ont servi à produire le nom de commande pour que \TeX l'identifie comme une seule unité lexicale et donc le prenne comme ce qui doit être montré :

```
\expandafter\show\csname \string\lvec \endcsname
```

Techniquement, un nom de commande est produit à partir des unités lexicales comprises entre `\csname` et `\endcsname`. À l'intérieur de cette construction, la commande `\string` transforme la commande `\lvec` en une suite de caractères débutant par une contre-oblique qui n'indique plus le début d'une commande. C'est pourquoi le nom de commande qui en résulte contient deux contre-obliques au départ. La commande `\expandafter` met en attente l'évaluation de la commande `\show` qui suit, afin que `\csname` puisse effectuer tout son travail avant que `\show` soit autorisé à regarder le résultat.

Il s'agit d'une assez belle brochette de commandes T_EX de bas niveau, mais après les avoir saisies, on est récompensé avec la sortie suivante :

```
> \lvec=\long macro:
[#1]#2->\ensuremath {#2_1+\cdots + #2_{#1}}.
<recently read> \lvec
```

Cette fois, on ne voit plus de ligne du fichier source après l'affichage de la commande, mais les mots `<recently read>`. Ils indiquent que T_EX a assemblé l'unité lexicale `\lvec` quelque part en mémoire, au lieu de la lire directement dans un fichier.

Que se passe-t-il lorsqu'on oublie le `\expandafter` initial dans l'instruction précédente ? On obtient le résultat :

```
> \csname=\csname.
1.5 \show\csname
          \string\lvec \endcsname
?
! Extra \endcsname.
1.5 \show\csname \string\lvec \endcsname
?
?
```

La première information indique que `\csname` est un `\csname`, ce qui ressemble beaucoup à un renseignement totalement inutile mais, en réalité, elle indique que `\csname` est une primitive ou un registre prédéfini du programme T_EX (contrairement à `\lvec`, par exemple, qui était un macro définie à partir de `\newcommand`). L^AT_EX montre également à quel niveau il est arrivé dans la ligne du source, en plaçant les unités lexicales non lues (`\string` et compagnie) au niveau de la ligne suivante. Dès la reprise, T_EX s'arrête de nouveau (en ayant consommé la ligne entière) pour se plaindre d'un `\endcsname` parasite, puisque le `\csname` correspondant a été montré mais non exécuté.

Détecter une primitive

La commande `\show` est utile pour obtenir des informations sur des commandes et leurs définitions, ou pour savoir si une commande est une primitive de T_EX. Elle n'aide pas à connaître les valeurs des registres de longueur ou de compteur. Par exemple,

Afficher les valeurs de registres

```
\show\parskip \show\topmargin \show\topsep
```

donne les informations suivantes :

```
> \parskip=\parskip.
1.5 \show\parskip
          \show\topmargin \show\topsep
?
?
```

```

> \topmargin=\dimen73.
1.5 \show\parskip \show\topmargin
                                \show\topsep
?
> \topsep=\skip23.
1.5 \show\parskip \show\topmargin \show\topsep

```

Avec ce que l'on a vu précédemment, on peut en déduire que `\parskip` est une primitive de \TeX (le fait qu'il s'agisse d'un ressort n'est pas indiqué), que `\topmargin` est en réalité le registre `\dimen` (longueur fixe) de numéro 73, et que `\topsep` est le registre `\skip` (ressort ou longueur élastique) de numéro 23.

Si l'on veut connaître la valeur de chacun de ces registres, il faut employer une autre primitive de \TeX , appelée `\showthe`, qui donne alors la sortie suivante sur le terminal et qui prouve également que `\parskip` est effectivement un ressort :

```

> 0.0pt plus 1.0pt.
1.5 \showthe\parskip

```

En utilisant `\showthe` de cette façon, on peut afficher les valeurs des registres de longueur alloués par `\newlength`, ainsi que les registres internes de \TeX tels que `\baselineskip` et `\tolerance`. En revanche, il n'est pas possible d'afficher les valeurs des compteurs \TeX , alloués par `\newcounter`, avec cette procédure. Pour cela, il faut utiliser une commande `\value` qui transforme un compteur \TeX en une forme qui est acceptée par `\showthe`. Par exemple,

```
\showthe\value{footnote}
```

affiche la valeur en cours du compteur de note de bas de page sur le terminal.

Composer les définitions de commandes et les valeurs de registres Au lieu d'afficher la signification d'une macro ou la valeur d'un registre au niveau du terminal, on peut composer ce type de données en utilisant la commande `\meaning` à la place de `\show`, et `\the` à la place de `\showthe`. La sortie est légèrement différente : le nom de l'unité lexicale n'est pas indiqué par `\meaning` ; seuls son type et sa signification (`meaning`) sont présentés. On peut comparer l'exemple qui suit avec la sortie montrée précédemment dans cette section.

<pre> \long macro:#1->\ensuremath {x_1,\ldots ,x_{#1}} 0.0pt plus 1.0pt 16.0pt 8.0pt plus 2.0pt minus 4.0pt footnote=0 </pre>	<pre> \newcommand\xvec[1]{\ensuremath{x_1,\ldots,x_{#1}}} \ttfamily % compose en typewriter \raggedright \meaning\xvec \par \the\parskip\par \the\topmargin \par \the\topsep \par footnote=\the\value{footnote} </pre>	B-3-1
--	--	-------

Si l'affichage des définitions de commandes ou des valeurs de registres n'est pas suffisant pour déterminer la raison d'un problème, on peut également tracer le comportement des commandes en action (voir section B.3.5 page 963).

B.3.2 Diagnostiquer les problèmes de coupures de pages

De temps en temps, L^AT_EX effectue des coupures de pages non souhaitées ou montre d'étranges blancs verticaux, et on voudrait comprendre d'où ils viennent et quelles sont précisément les dimensions utilisées. Pour cela, T_EX offre quelques outils de traçage de bas niveau.

Affichage symbolique du contenu de pages

En spécifiant `\showoutput` quelque part dans le document, T_EX affiche (en commençant par la page en cours) les représentations symboliques des pages complètes sur le terminal et dans le fichier de trace. Cela produit une grande quantité d'informations. On ne montre qu'un court extrait obtenu en ayant compilé le premier paragraphe de cette section séparément.

Chaque page de sortie commence par le texte `Completed box being shipped out`, suivi du numéro de page en cours entre crochets. On a alors un grand nombre de lignes montrant les boîtes qui construisent la page. Cela débute par une `\vbox` (boîte verticale) et sa taille en pt contenant la page entière. Pour indiquer que quelque chose est le contenu de la boîte, tout ce qu'elle contient est mis en retrait récursivement en utilisant des points au lieu d'espaces. Les espaces, même s'il s'agit de longueurs rigides, sont indiqués par le mot-clé `\glue` (voir lignes 3 et 6) ; les espaces élastiques (ressorts) ont des composantes `plus` et `minus` en plus. Par exemple, la `\glue` de 16.0pt de la ligne 3 est un espace vertical qui vient de `\topmargin` (voir également l'exemple B-3-1 page précédente). Dans cet extrait, on peut aussi voir une `\vbox` vide de hauteur 12pt (lignes 5 à 7), qui est l'en-tête, suivie en ligne 8 de l'espace de `\headsep` (25pt), suivie de la boîte contenant le rectangle d'empagement et qui commence à la ligne 10. Les lignes 15 et suivantes montrent la façon dont chaque caractère s'affiche ; ici, `\T1/cmr/m/n/10` indique la fonte des caractères. La `\glue` entre eux (par exemple, en ligne 17) est un espace intermot avec ses composantes d'étirement et de compression. On peut remarquer la façon de spécifier la ligature « fi » en ligne 24.

```

1 Completed box being shipped out [1]
2 \vbox(621.0+0.0)x407.0
3  .\glue 16.0
4  .\vbox(605.0+0.0)x345.0, shifted 62.0
5  ..\vbox(12.0+0.0)x345.0, glue set 12.0fil
6  ... \glue 0.0 plus 1.0fil
7  ... \hbox(0.0+0.0)x345.0
8  .. \glue 25.0
9  .. \glue(\lineskip) 0.0
10 .. \vbox(538.0+0.0)x345.0, glue set 478.00497fil
11 ... \write-{}
12 ... \glue(\topskip) 2.50183
13 ... \hbox(7.49817+2.49939)x345.0, glue set 0.36246
14 .... \hbox(0.0+0.0)x15.0
15 .... \T1/cmr/m/n/10 E
16 .... \T1/cmr/m/n/10 n

```

```

17 ....\glue 3.33252 plus 1.66626 minus 1.11084
18 ....\T1/cmr/m/n/10 s
19 ....\T1/cmr/m/n/10 p
20 ....\kern0.27771
21 ....\T1/cmr/m/n/10 é
22 ....\T1/cmr/m/n/10 c
23 ....\T1/cmr/m/n/10 i
24 ....\T1/cmr/m/n/10 ^^ (ligature fi)
25 ....\T1/cmr/m/n/10 a
26 ....\T1/cmr/m/n/10 n

```

Le deuxième exemple de tracé de page montre l’affichage symbolique des structures proches d’une coupure de ligne. On voit l’espace ajouté par T_EX à droite de la ligne de texte (`\rightskip` de la ligne 4) et la boîte contenant la ligne. La ligne 5 a alors un retrait qui diminue (pour montrer que la boîte de la ligne de texte en cours est terminée). Elle contient une représentation symbolique du coût d’une coupure de page à ce niveau. La valeur affichée est celle du paramètre `\clubpenalty`¹. Cette pénalité est suivie, en ligne 6 d’un espace vertical ajouté entre les lignes et calculé par T_EX en prenant la valeur de `\baselineskip` et en lui soustrayant la profondeur de la ligne précédente et la hauteur de la boîte de la ligne suivante, qui commence en ligne 7.

```

1 ....\T1/cmr/m/n/10 (
2 ....\T1/cmr/m/n/10 e
3 ....\T1/cmr/m/n/10 n
4 ....\glue(\rightskip) 0.0
5 ..\penalty 150
6 ..\glue(\baselineskip) 2.00244
7 ...\hbox(7.49817+2.49939)x345.0, glue set - 0.30447
8 ....\T1/cmr/m/n/10 c
9 ....\T1/cmr/m/n/10 o

```

Le dernier exemple montre l’affichage symbolique d’un début de page produit par :

```
\begin{itemize} \item test \end{itemize} \section{Test}
```

La partie intéressante est celle produite par le `\end{itemize}` et le `\section{Test}`. On voit (lignes 1 à 7) une curieuse collection d’instructions `\glue` : la plupart d’entre-elles s’annulent les unes les autres et sont mélangées à un certain nombre de `\penalty` :

¹La pénalité d’une coupure après la première ligne d’un paragraphe (ligne veuve) est donnée par le paramètre `\clubpenalty`; le coût d’une coupure avant la dernière ligne (ligne orpheline) est donné par le paramètre `\widowpenalty`. Ces deux paramètres ont une valeur par défaut de 150, ce qui décourage sensiblement une coupure à ces emplacements.

```

1 ...\penalty -51
2 ...\glue 10.0 plus 3.0 minus 5.0
3 ...\glue -10.0 plus -3.0 minus -5.0
4 ...\penalty -300
5 ...\glue 10.0 plus 3.0 minus 5.0
6 ...\glue -10.0 plus -3.0 minus -5.0
7 ...\glue 15.0694 plus 4.30554 minus 0.86108
8 ...\glue(\parskip) 0.0 plus 1.0
9 ...\glue(\baselineskip) 8.12001
10 ...\hbox(9.87999+0.0)x345.0, glue set 290.70172fil

```

Ces lignes sont dues à plusieurs commandes `\addpenalty` et `\addvspace`. Par exemple, les lignes 1 et 2 sont la pénalité et l'espace fixe ajoutés par `\end{itemize}`. La commande `\section` ajoute alors un point de coupure pour indiquer que cet emplacement est un bon endroit pour une coupure de page (en utilisant `\@secpenalty` qui a une valeur de `-300`). En réalité, cette coupure devrait se faire avant la `\glue` de la ligne 2 mais, comme il est impossible de retirer le matériel de la réserve verticale, `\addpenalty` utilise une astuce qui consiste à ajouter un espace négatif (ligne 3), à ajouter la pénalité (ligne 4), pour enfin réintroduire la `\glue` (ligne 5). La même méthode est utilisée en lignes 6 et 7 par `\addvspace` pour ajouter l'espace vertical qui précède le titre.

T_EX ajoute les lignes 8 et 9 lorsqu'il place le texte du titre (ligne 10) dans la réserve. On peut noter que, techniquement, le titre est considéré comme un « paragraphe », donc on ajoute `\parskip`. C'est la raison pour laquelle augmenter ce paramètre nécessite quelques précautions. La même prudence doit être de mise lorsqu'on souhaite modifier d'autres paramètres (tels que celui ajouté en ligne 7).

La commande `\showoutput` produira également des affichages symboliques pour les débordements de boîtes. L'affichage se terminera à la prochaine accolade fermante ou à la prochaine balise de fin d'environnement. Par conséquent, si l'on veut voir la sortie de pages complètes, on doit s'assurer que la coupure de page survient avant que le prochain groupe soit terminé.

*Effet de bord de
`\showoutput`*

Tracer les décisions de coupures de pages

Pour tracer les décisions de coupures de pages, T_EX permet l'affichage d'informations symboliques qui peuvent être activées en assignant une valeur strictement positive au compteur interne `\tracingpages` :

```
\tracingonline=1 \tracingpages=1
```

Le fait d'assigner une valeur strictement positive à `\tracingonline` permet d'afficher les informations non seulement dans le fichier de trace mais également au niveau du terminal.

En traitant le précédent paragraphe qui commence par « Pour tracer les... » en tant que document séparé, on obtient les lignes d'informations de trace suivantes :

```

1 %% goal height=522.0, max depth=4.0
2 % t=10.0 g=522.0 b=10000 p=150 c=100000#

```

```

3 % t=22.0 g=522.0 b=10000 p=150 c=100000#
4 % t=55.0 plus 4.0 g=522.0 b=10000 p=-51 c=100000#
5 % t=77.0 plus 8.0 g=522.0 b=10000 p=300 c=100000#
6 % t=89.0 plus 8.0 g=522.0 b=10000 p=0 c=100000#
7 % t=90.94397 plus 8.0 plus 1.0fil g=522.0 b=0 p=-10000 c=-10000#

```

La première ligne, débutant par deux signes «%», montre la hauteur à atteindre pour la page (c'est-à-dire 522pt ici), ce qui signifie 43 lignes avec un `\baselineskip` de 12pt en oubliant 2pt, puisque le saut qui positionne la première ligne de base (`\topskip`) a une valeur de 10pt. Si la hauteur à atteindre ne correspond pas à un nombre entier de fois le nombre de lignes, il risque de se produire des problèmes de boîtes verticales incomplètes (`underfull \vbox`).

*Longueur à atteindre
pour une coupure*

Les lignes restantes, débutant par un unique signe «%», indiquent une position potentielle de coupure que T_EX a envisagée. On peut interpréter ces lignes de la façon suivante : `t=` montre la hauteur atteinte dans la réserve verticale et, si cette réserve contient des ressorts verticaux, la valeur totale de leur extensibilité et compressibilité. La ligne 4, par exemple, montre que dans ce livre, les verbatim hors-texte ont un espace supplémentaire de 10pt, plus une extensibilité de 4pt (les lignes verbatim sont composées dans une fonte plus petite avec une `\baselineskip` de seulement 11pt) et la même quantité est ajoutée entre les lignes 4 et 5.

*But de hauteur de
page*

Le `g=` spécifie la hauteur à atteindre à cet emplacement. Cette valeur n'est modifiée que lorsque des objets, tels que des flottants, ont réduit l'espace disponible dans la réserve.

Médiocrité de page

Avec `b=`, T_EX indique la médiocrité de la page si la coupure devait se faire à ce niveau. La médiocrité est calculée avec le facteur par lequel l'étirement ou la compression dans `t=` doit être multipliée pour atteindre la hauteur cible indiquée avec `g=`. Dans cet exemple, la page n'est pas tout à fait remplie et la médiocrité vaut donc toujours 10000 (infiniment mauvais), sauf à la ligne 7 où, à cause de l'ajout d'une extensibilité de `fil`, la page est soudainement considérée comme optimale (`b=0`).

Pénalité de coupure

À chaque point de coupure, T_EX associe une valeur numérique `\penalty` qui indique le coût d'une coupure à cet emplacement. Sa valeur est donnée par `p=`. Par exemple, il est interdit de couper directement avant le verbatim hors-texte. C'est pourquoi il y a une augmentation forte de `t=` entre les lignes 3 et 4. En revanche, une coupure après le hors-texte donne un bonus (`p=-51`). La ligne 5 montre qu'une coupure de ligne après la première ligne d'un paragraphe de deux lignes qui suit un hors-texte est considérée comme mauvaise (`p=300`), puisqu'il en résulterait à la fois une ligne veuve et une ligne orpheline : les valeurs de `\clubpenalty` et de `\widowpenalty` sont chacune égales à 150 et ces valeurs sont ajoutées.

*Coût d'une coupure
de page*

Finalement, `c=` indique le coût d'une coupure à ce niveau, qui est calculé à partir de la médiocrité de la page résultante (`b=`) et de la pénalité de coupure à ce niveau (`p=`). T_EX regarde les valeurs de ces coûts et coupera éventuellement au niveau de la coupure de moindre coût. Si la ligne se termine par `#`, T_EX pense qu'il s'agit du meilleur emplacement vu jusqu'ici. Dans l'exemple, toutes les lignes montrent ce `#` – ce qui n'est pas surprenant, puisque T_EX considère que toutes ces coupures sauf la dernière sont aussi mauvaises les unes que les autres.

Si une page devient trop pleine en cas de coupure à un certain niveau, T_EX l'indique avec `b=*`. À ce niveau, T_EX arrête de regarder d'autres points de coupures et coupe la page au meilleur emplacement vu jusqu'ici.

Pour plus de détails sur ce type d'affichage, on pourra consulter [87, p.112].

B.3.3 Diagnostiquer et résoudre les problèmes de coupures de lignes

Si T_EX n'est pas capable de trouver un ensemble convenable d'emplacements où couper un paragraphe en lignes, il produira en dernier ressort une ou plusieurs lignes qui débordent dans la marge (`overfull`). Pour chacune d'entre elles, on obtient un avertissement à l'écran et dans le fichier de trace, comme

```
Overfull \hbox (21.30931pt too wide) in paragraph at lines 4257--4272
/futr9e/qui montre l'af-fi-chage sym-bo-lique de la ligne de texte et
des nu-mé-ros de lignes du paragraphe|
```

qui montre l'affichage symbolique de la ligne de texte et des numéros de lignes du paragraphe qui la contient. En regardant cet affichage symbolique, on peut facilement s'apercevoir que le problème était que T_EX était incapable¹ de couper le mot « paragraphe ». Pour repérer facilement ces lignes, on peut assigner une longueur strictement positive au paramètre `\overfullrule`. Pour ce paragraphe, il vaut 5pt, ce qui produit un rectangle noir indiquant clairement la ligne qui déborde. Les classes standard de document permettent ce comportement avec l'option `draft`. D'un autre côté, il peut être embêtant de noter de la sorte les lignes qui ne débordent que très légèrement. Pour l'éviter, il suffit de modifier le paramètre `\hfuzz` (égal à 0.1pt par défaut) ; seules les lignes débordant de plus de cette valeur seront indiquées.

Si T_EX est incapable de couper un paragraphe de façon satisfaisante, la raison la plus probable est une impossibilité de couper un mot avec les valeurs des paramètres en cours pour l'algorithme de construction de paragraphe. Cela peut aussi être dû à des colonnes trop étroites avec un ensemble de valeurs de paramètres qui n'autorisent pas, pour des raisons esthétiques, une certaine souplesse dans l'étirement des lignes. Dans ce dernier cas, le seul remède est, en général, de modifier partiellement le texte.

Gérer les problèmes de coupure de mot

Avec les motifs de césures adéquats chargés, T_EX est capable de réaliser un assez bon travail pour de nombreuses langues (voir [123]). Cependant, il ne trouve pas forcément tous les emplacements de coupures potentiels et il a donc parfois besoin d'être aidé dans cette tâche. Pour savoir quels sont les points de coupures trouvés par T_EX dans des mots ou des phrases, par exemple « catastrophe », on peut placer ces mots ou ces phrases dans l'argument de la commande `\showhyphens` :

```
\showhyphens{catastrophe antithèse}
```

¹En réalité, T_EX est tout à fait capable de couper pa-ra-graphe ; pour l'exemple, nous l'avons explicitement interdit. Le paragraphe aurait été parfait sans cette intervention !

Le traitement de cette instruction par \TeX indique, sur le terminal et dans le fichier de trace, les points de coupures potentiels sur ces mots. Ces derniers sont indiqués par un trait d'union :

```
[] \T1/cmr/m/n/10 catas-tro-phe an-tithèse
```

Si l'on veut ajouter les points de coupures oubliés ou bien les modifier, on peut les spécifier localement pour un mot dans le texte en utilisant `\-`, par exemple :

```
cata\-\stro\-\phe an\-\ti\-\thèse
```

On peut également utiliser une déclaration `\hyphenation` dans le préambule :

```
\hyphenation{cata-stro-phe an-ti-thèse}
```

Cette dernière technique est particulièrement utile lorsqu'on a détecté une mauvaise coupure ou que l'on utilise fréquemment un mot pour lequel on sait que \TeX oublie des points de coupures importants. On peut noter que de telles spécifications explicites portent exactement sur les mots indiqués. Ainsi, le pluriel « catastrophes » ne sera pas modifié, à moins de spécifier ses coupures également.

Les déclarations `\hyphenation` s'appliquent à la langue en cours. Lorsqu'un document utilise plusieurs langues – par exemple en utilisant les méthodes fournies par le système `babel` – on doit alors passer dans la bonne langue avant d'utiliser ce type de déclarations.

Tracer l'algorithme de coupures de paragraphes

Comme \TeX utilise un algorithme global pour optimiser les coupures de lignes d'un paragraphe, il n'est pas toujours facile de comprendre pourquoi il a choisi telle solution plutôt que telle autre. En cas de besoin, on peut tracer les décisions de coupures de lignes en utilisant la déclaration suivante¹ :

```
\tracingparagraphs=1 \tracingonline=1
```

Pour le lecteur qui souhaite réellement comprendre les raisons qui déterminent certaines décisions, on montre un exemple de données avec des explications détaillées.

La trace de l'algorithme produit une sortie assez effrayante. Par exemple, l'un des paragraphes précédents à produit des données qui commencent comme suit :

```
1 @firstpass
2 @secondpass
3 []/futr9e/Les dé-cla-ra-tions [] s'ap-liquent à la langue en cours. Lors-qu'un
4 @ via @@0 b=1024 p=0 d=1079156
```

Jusqu'à trois passes sur le paragraphe La ligne 2 indique que \TeX a immédiatement abandonné un essai de coupure de lignes sans coupure de mots. Cela est dû à la valeur de `\pretolerance` fixée à 100 dans les sources de ce livre ; sinon, \TeX aurait pu éventuellement réaliser les coupures de lignes sans coupures de mots (dans

¹Ces paramètres sont également activés par une commande `\tracingall`, mais on obtient alors un grand nombre de données de trace, même si l'on est intéressé par quelque chose de totalement différent.

un texte anglais, une grande proportion de paragraphes peuvent être composés sans césures¹). En plus d'une @secondpass, on peut parfois voir une @emergencypass, qui indique que même avec des césures, il a été impossible de trouver une solution à la coupure des lignes d'un paragraphe; une autre passe utilisant \emergencystretch a alors été essayée². La ligne 3 montre ce que T_EX a lu avant de trouver la première fin de ligne potentielle qui donne une médiocrité inférieure à $\infty = 10000$. La ligne 4 donne des détails sur cette coupure potentielle. Ces lignes débutent avec un unique @; le via donne le point de coupure précédent (dans ce cas @@0, qui indique le début du paragraphe), la médiocrité de la ligne (b=), la pénalité de coupure à cet emplacement (p=) et qu'on appelle le démérite (d=) associé à cette coupure (il s'agit du coût qui prend en compte la médiocrité, la pénalité de quelques informations, comme le fait de couper au niveau d'un trait d'union, et une compatibilité visuelle avec la ligne précédente).

```
5 @01: line 1.0 t=1079156 -> @@0
```

À la ligne 5, T_EX indique qu'il a été possible de former une ligne très lâche qui se termine au niveau du point de coupure montré en ligne 3, avec un coût total (t=) égal au démérite montré en ligne 4. Cette ligne commencerait au point de coupure @@0. La notation line 1.0 donne le numéro de ligne en train d'être construite, et les suffixes .3, .2, .1 et .0 indiquent des espaces intermots respectivement très lâches, lâches, décentes et serrées dans la ligne. Cette classification sert lorsque T_EX compare les lignes consécutives pour voir si elles sont compatibles.

T_EX trouve maintenant de plus en plus de points de coupures potentiels, comme après « do- » en ligne 6 et après « cu- » en ligne 9. À chaque fois, T_EX indique le type de la ligne qui sera formée s'il coupe à cet emplacement. Si un b=* apparaissait quelque part dans les données de trace, cela signifierait que T_EX ne peut pas trouver de point de coupure convenable pour une ligne et qu'il doit choisir une solution interdite (c'est-à-dire qui dépasse \tolerance pour cette ligne).

```
6 do-
7 @\discretionary via @@0 b=57 p=50 d=6989
8 @02: line 1.1- t=6989 -> @@0
9 cu-
10 @\discretionary via @@0 b=0 p=50 d=2600
11 @03: line 1.2- t=2600 -> @@0
```

En coupant le mot document, T_EX trouve donc deux points de coupures supplémentaires. Cette fois, on peut voir (lignes 7 et 10) une pénalité de 50 – la valeur du paramètre \hyphenpenalty (coupure de ligne après un trait d'union) – attachée à cette coupure. La ligne 9 est le dernier point de coupure qui peut être effectué sur la première ligne de ce paragraphe. Tous les autres points de coupures produiraient une ligne qui déborderait. La prochaine ligne de trace montre de nouveau plus de texte. Aucune des coupures de mots potentielles ne peut être utilisée ici puisqu'il en résulterait une ligne dépassant \tolerance.

```
12 12 ment uti-lise plu-sieurs langues -- par exemple en uti-li-sant les mé-thodes four-
13 13 @\discretionary via @01 b=914 p=50 d=856276
14 14 @\discretionary via @02 b=2393 p=50 d=5786909
15 15 @\discretionary via @03 b=4765 p=50 d=22823125
16 16 @04: line 2.0- t=1935432 -> @01
```

Ici, le point de coupure pour former la deuxième ligne peut être placé de trois façons différentes : en commençant à partir du point de coupure @01 (ligne 13), du point de coupure @02 (ligne 14)

¹Pour le L^AT_EX Companion avec ces noms de commandes très longs, c'est plus difficile. Cela devient encore moins facile avec la traduction française [NdT].

²Pour que cela puisse arriver, il faut que \emergencystretch ait une valeur strictement positive. Voir sa présentation à la section 3.1.11.

ou du point de coupure @3 (ligne 15). Si l'on ne compare que ces trois solutions pour former la deuxième ligne, alors la première solution semble préférable : elle a une médiocrité de 914 tandis que les deux autres ont des médiocrités de 2393 et 4765, ce qui entraîne des coûts ($t=$) de 7 et 27 fois supérieurs environ. Ici, en quelque sorte, nous avons de la chance : la meilleure coupure pour la première ligne entraîne la meilleure solution pour la seconde ligne mais ce n'est évidemment pas obligatoire et T_EX ne choisira pas nécessairement cette solution. Pour le choix définitif, il doit ajouter les coûts de toutes les lignes. T_EX enregistre en ligne 16 que pour atteindre le point de coupure indiqué en ligne 13, il faut passer par @1 et que le coût total est de 1935432. Pour le reste du traitement, T_EX n'a pas besoin de savoir qu'il existe plusieurs façons d'atteindre @4. Il a juste à mémoriser la meilleure façon d'y arriver.

Plus précisément, T_EX doit enregistrer la meilleure façon d'atteindre un point de coupure pour chacun des quatre types de lignes (très lâche, lâche, décente, serrée), puisque l'algorithme attache des démérites différents si des lignes adjacentes sont visuellement incompatibles (par exemple, une ligne lâche suivie d'une ligne serrée). Plus loin dans l'affichage (les lignes 17 à 57 ne sont pas montrées), on obtient la sortie suivante :

```

41 58 de
42 59 @ via @6 b=2 p=0 d=144
43 60 @ via @7 b=46 p=0 d=3136
44 61 @@14: line 3.1 t=6177 -> @7
45 62 @@15: line 3.2 t=7574 -> @6

```

Cette sortie indique qu'il y a deux façons de former une ligne se qui se termine par « de » : en commençant à partir de @6 ou à partir de @7. Une ligne de médiocrité inférieure ou égale à 12 est considérée comme décente (suffixe .2) ; une ligne s'étirant avec une médiocrité strictement inférieure à 100 est considérée comme lâche (suffixe .1). On peut voir ici que T_EX mémorise deux points de coupures faisables : un en passant par @6, et un autre passant par @7.

Le chemin de points de coupures qui sera finalement sélectionné n'est décidé qu'en toute fin de travail. Ainsi, toute modification, même mineure, peut amener T_EX à choisir un ensemble de points de coupures totalement différent pour construire la meilleure solution au problème de coupure du paragraphe en lignes, car cet ensemble produirait alors le coût total le plus faible. En raison de la complexité de l'algorithme, des modifications mineures ont parfois des résultats surprenants. Par exemple, la suppression d'un mot peut donner un paragraphe avec une ligne de plus. Cela peut arriver, car T_EX décide alors d'utiliser des lignes lâches de façon plus uniforme ou bien d'éviter une coupure de mot. Pour plus de détails sur tous les paramètres qui influencent les décisions de coupures de lignes, on peut se référer à [87, p.98]. En cas de besoin, on peut forcer des points de coupures à certains emplacements avec `\linebreak`, ou les empêcher avec `\nolinebreak`, ou en utilisant `~` à la place d'un espace. Il est clair que les choix en début de paragraphe sont assez limités et qu'il faudra éventuellement réécrire une phrase pour éviter une mauvaise coupure. En revanche, plus en avant dans le paragraphe, à peu près toutes les coupures potentielles deviennent faisables sans dépasser la `\tolerance` spécifiée.

Allonger ou raccourcir un paragraphe

Le compteur interne `\looseness` est un autre outil de bas niveau que l'on peut utiliser pour corriger ponctuellement la forme finale d'un paragraphe. En lui donnant une valeur non nulle n , T_EX essaie d'allonger le prochain paragraphe de n lignes (n positif) ou de le raccourcir de $-n$ lignes (n négatif), tout en gardant toutes les autres conditions

(par exemple, la `\tolerance` autorisée). Ainsi, le dernier paragraphe de la section qui précède a été allongé artificiellement en le saisissant sous la forme :

```
\looseness=1
Le chemin de points de coupures qui sera finalement sélectionné
```

Modifier la valeur de `\looseness` ne garantit pas son effet. Il est plus difficile de raccourcir un paragraphe que de l'allonger, puisque les espaces intermots ont une compression plus faible que leur étirement autorisé normal. Les meilleurs résultats sont obtenus sur des paragraphes longs qui ont une dernière ligne courte. Inversement, étendre un paragraphe fonctionne mieux sur des paragraphes longs qui ont une dernière ligne presque pleine, bien qu'il soit nécessaire de placer les derniers mots du paragraphe à l'intérieur d'une `\mbox` pour s'assurer qu'il y aura plus d'un mot sur la dernière ligne. Dans le dernier paragraphe de la section précédente, seul le mot « spécifiée » a été placé dans une `\mbox` ; sans cette précaution, il y aurait eu la coupure « spéci-fiée ».

B.3.4 Autres outils de bas niveau de trace

T_EX offre d'autres commandes et paramètres entiers internes qui peuvent parfois aider à déterminer la source d'un problème. Ils sont énumérés ici, accompagnés d'une brève explication sur leur utilisation.

On a déjà vu `\tracingonline`. En lui assignant une valeur strictement positive, toutes les informations sont montrées sur le terminal ; sinon, la plupart d'entre elles sont uniquement écrites dans le fichier de trace. Ce paramètre est automatiquement activé par `\tracingall`. *Trace en ligne*

Avec `\tracingoutput`, la trace de contenu des pages est activée. Deux paramètres supplémentaires contrôlent ce qui est montré : `\showboxdepth` (jusqu'à quel niveau d'emboîtement affiche-t-on) et `\showboxbreadth` (quelle quantité de matériel va-t-on montrer pour chaque niveau). Tout ce qui va dépasser ces valeurs s'affichera sous la forme raccourcie `etc.` ou `[]` (qui indique une boîte à l'intérieur d'une autre) au niveau de l'affichage symbolique.

Pour voir le contenu d'une boîte produite avec `\sbox` ou `\savebox`, on peut utiliser la commande T_EX `\showbox` : *Contenu des boîtes*

```
\newsavebox\test\sbox\test{Un test} {\tracingonline=1\showbox\test}
```

Cependant, le résultat serait totalement inutile si l'on n'avait pas modifié à la fois `\showboxdepth` et `\showboxbreadth` en même temps. Une meilleure stratégie est d'utiliser la commande `\showoutput` de L^AT_EX. Celle-ci met ces deux paramètres à leur valeur maximale et `\tracingoutput` à la valeur 1, de façon à obtenir l'information la plus détaillée possible. Cette commande est appelée par `\tracingall`. La ligne d'appel précédente peut alors être remplacée par la forme :

```
{\showoutput\showbox\test}
```

On peut noter l'utilisation des accolades pour limiter la portée de `\showoutput`. Sans elles, on aurait vu la description des boîtes suivantes de la page, ce qui n'aurait sans doute pas présenté beaucoup d'intérêt. Le même type d'affichage symbolique que celui étudié à la section B.3.2 est envoyé sur le terminal :

```
> \box26=
\hbox(6.83331+0.0)x32.5556
.\OT1/cmr/m/n/10 U
.\OT1/cmr/m/n/10 n
.\glue 3.33333 plus 1.66666 minus 1.11111
.\OT1/cmr/m/n/10 t
.\OT1/cmr/m/n/10 e
.\OT1/cmr/m/n/10 s
.\OT1/cmr/m/n/10 t
```

En ajoutant `\scrollmode` ou `\batchmode` avant la commande `\showbox`, on indique à \TeX de ne pas s'arrêter à ce niveau. On peut alors étudier le fichier de trace.

Restaurations locales

Pour voir quelles sont les valeurs et les définitions que \TeX restaure lorsqu'un groupe se termine, on assigne une valeur strictement positive à `\tracingrestores`. Elle est automatiquement activée par `\tracingall`.

Pile des listes de \TeX

En utilisant `\showlists`, on demande à \TeX d'afficher la pile des listes (verticales et horizontales) sur lesquelles il est en train de travailler. Par exemple, en plaçant `\showlists` dans la note de bas de page¹ de ce paragraphe, on obtient la sortie suivante dans le fichier de trace :

```
### horizontal mode entered at line 4661 (language12:hyphenmin2,3) []
spacefactor 1000, current language 12
### internal vertical mode entered at line 4661
prevdepth ignored
### horizontal mode entered at line 4650 (language12:hyphenmin2,3) []
spacefactor 1000, current language 12
### vertical mode entered at line 0
### current page: []
total height 469.22221 plus 24.0
goal height 538.0
```

Ici, le texte de la note commençait à la ligne 4661 et on a assigné une valeur de 1000 à `\spacefactor` au départ. La note elle-même a commencé sur la même ligne, contribuant au « mode vertical interne », et \TeX n'a fort justement pas tenu compte de la valeur extérieure de `\prevdepth`. La note faisait partie d'un paragraphe qui a commencé à la ligne 4650 qui, à son tour, faisait partie d'une liste verticale commencée à la ligne 0, ce qui indique la réserve verticale principale. Enfin, la sortie donne quelques informations sur la liste de la page en cours, avec sa hauteur actuelle, son but en hauteur et la valeur de `\prevdepth` (c'est-à-dire la profondeur de la dernière ligne de la page à ce moment). Pour la version française, la sortie montre également que le texte principal et la note

¹Une note crée une nouvelle liste verticale et, à l'intérieur, une liste horizontale pour le texte.

sont composés en utilisant la langue numéro 12, et que les coupures de mots doivent laisser au moins deux lettres à gauche et au moins trois à droite [NdT].

En raison des valeurs par défaut de `\showboxbreadth` et `\showboxdepth`, les contenus de toutes ces listes sont raccourcis en `[]`. Pour obtenir plus de détails, il faut modifier ces paramètres ou utiliser `\showoutput\showlists`.

La commande `\tracingcommands` employée seule est peu utile, mais elle le devient lorsqu'elle est utilisée avec d'autres options de traçage. Elle montre toutes les primitives utilisées par T_EX au cours de la compilation. Une commande fréquemment associée est le paramètre entier interne `\tracingmacros`, qui montre tous les développements de macros effectués par T_EX. En lui assignant la valeur 2, il montre également le développement des structures conditionnelles. Ces deux paramètres sont automatiquement activés par `\tracingall`.

Suivre la compilation

Lorsque tout se passe bien, il y a peu de chance que T_EX accède à une position de la fonte en cours qui ne correspond à aucun glyphe. Cependant, certaines commandes, comme `\symbol`, peuvent demander explicitement n'importe quel numéro de caractère. Malheureusement, T_EX ne considère pas les caractères manquants comme une erreur (ce qu'il devrait) ; il ne fait que reporter ces caractères manquants dans le fichier de trace par un message anodin, et il ne procède à cet avertissement que si `\tracinglostchars` a une valeur strictement positive. L^AT_EX essaie d'être obligeant en initialisant ce paramètre interne à 1.

*Tracer les caractères
inexistants*

Finalement, on peut demander à T_EX de traiter les fichiers sources ligne par ligne. En assignant la valeur 1 à `\pausing`, chaque ligne du source est d'abord affichée (précédée d'un =>). T_EX attend alors des instructions sans traiter cette ligne. Si l'on tape la touche *Entrée*, T_EX utilise la ligne sans rien changer ; toute autre saisie indique que T_EX doit utiliser ce qui a été entré au clavier au lieu de la ligne affichée. T_EX développe et compose alors ce qui lui a été passé, affiche la ligne suivante, et s'arrête de nouveau. Pour continuer normalement, on peut répondre `\pausing=0` sans oublier que cela sera utilisé à la place de la ligne courante et qu'il faudra sans doute saisir toute la ligne avant de spécifier cette instruction.

*Traiter un document
pas à pas*

B.3.5 Exécuter des commandes de trace de façon sélective – trace

La commande L^AT_EX `\tracingall` (héritée de Plain T_EX) permet d'activer toutes les fonctions de trace. Il y a cependant quelques problèmes avec cette commande :

1. Il n'y a pas de commande correspondante pour tout désactiver. En conséquence, il faut délimiter la portée, ce qui n'est pas toujours commode, ni même possible.
2. Certaines parties de L^AT_EX produisent d'énormes quantités de données de trace qui sont d'un intérêt limité, voire totalement inutiles pour le problème à traiter.

Par exemple, si L^AT_EX doit charger une nouvelle fonte, il appelle des routines internes de NFSS qui analysent les tables de définition de fonte, et effectuent d'autres tâches. Dans 99,9 % des cas, on n'est absolument pas intéressé par cette partie du traitement mais par les deux qui précèdent et les cinq qui suivent. Cependant, il va falloir examiner quelques centaines de lignes de sortie et essayer de localiser les lignes intéressantes.

Une instruction, telle que `\setlength\linewidth{1cm}`, est un autre exemple parlant. Avec \LaTeX standard, cela donne cinq lignes de sortie de trace. Mais si l'extension `calc` a été chargée, on arrive à une soixantaine de lignes, ce qui n'était sans doute pas voulu et pas vraiment utile, à moins de vouloir déboguer les routines d'analyse de `calc` (qui n'ont normalement pas besoin de l'être).

Pour résoudre le premier problème, l'extension `trace` (voir [130]), de Frank Mittelbach, définit la paire de commandes `\traceon` et `\traceoff`. Si \LaTeX est utilisé avec le moteur \TeX , alors `\traceon` est peu ou prou un autre nom pour `\tracingall` : il active les mêmes commandes de trace (quoique dans un ordre différent pour éviter d'avoir à se faire tracer lui-même). Si \LaTeX utilise le moteur $e\TeX$, alors la trace des instructions et des groupes est également activée¹.

*Plus d'informations
de trace avec $e\TeX$*

Une autre différence entre `\traceon` et `\tracingall` est que cette dernière affiche toujours les informations de trace sur le terminal, alors que l'on peut demander à `\traceon` de ne les écrire que dans le fichier de trace en spécifiant l'option `logonly`. C'est utile lorsque l'écriture sur le terminal est très lente (par exemple, lorsqu'on compile dans un buffer shell à l'intérieur d'`emacs`).

Pour résoudre le second problème, l'extension `trace` offre un certain nombre de commandes internes pour inactiver temporairement la trace. Elle redéfinit la plupart des fonctions d'affichage interne de \LaTeX afin de désactiver l'affichage lorsqu'elle sont exécutées. Par exemple, la fonction de chargement de fontes est gérée de cette façon. Si un document débute par les deux formules

```
$a \neq b$ \small $A = \mathcal{A}$
```

\LaTeX charge alors 22 nouvelles fontes² à cet emplacement. En utilisant le `\tracingall` standard sur cette ligne, on se retrouverait face à environ 7500 lignes de sortie sur le terminal. En revanche, si `\traceon` est utilisé, 350 lignes « seulement » seront produites (principalement pour tracer `\small`).

Les commandes pour lesquelles la trace est désactivée sont peu nombreuses et très certainement sans rapport avec le problème en cours. Cependant, s'il y a besoin de revenir à une trace complète, on peut soit utiliser `\tracingall`, soit spécifier l'option `full`. Dans ce dernier cas, `\traceon` va tout tracer, mais on peut toujours rediriger sa sortie vers le fichier de trace.

¹Les commandes $e\TeX$ correspondantes sont `\tracingassigns` et `\tracinggroups`; voir [28].

²On peut le vérifier avec l'option `loading` de l'extension `tracefmt`.