

Partie I.

KOMA-Script pour les auteurs

Dans cette partie, vous trouverez des informations pour les auteurs d'articles, de rapports et de livres. Il est supposé qu'un utilisateur moyen est peu intéressé par le fonctionnement de KOMA-Script et où se situent les difficultés de programmation. En outre pour cet utilisateur, les options obsolètes et les instructions qui en découlent sont de peu d'intérêt. Il préférera sans doute savoir comment réaliser un travail avec, peut-être, un intérêt pour la typographie.

Les passages qui contiennent des informations et des justifications supplémentaires, et ont donc moins d'intérêt pour le lecteur impatient, sont, dans l'ouvrage original, en police sans serif et peuvent être ignorés si nécessaire. Si vous êtes intéressés par plus d'informations sur la mise en œuvre, les effets secondaires avec d'autres packs, les options et instructions obsolètes, référez vous à la partie II du guide qui décrit les fonctions créées spécialement pour les auteurs de packs et de classes.

Je n'ai écarté aucune section, même si les chapitre 4 (la lettre `scrlltr2`), chapitre 7 (jour et heure avec `scrdate` et `scrtime`), chapitre 8 (l'accès aux fichiers d'adresses avec `scraddr`) et chapitre 9 (création de fichiers d'adresses à partir d'une base de données d'adresses) ne semblent pas, à mes yeux, d'un intérêt évident pour l'usage que j'espère faire de KOMA-Script. Ce n'est pas tout à fait exact puisque je n'ai pas repris le chapitre qui traite de la mise en page et des formats d'enveloppes destinés au Japon. Que les utilisateurs concernés par ce chapitre m'en excusent et se réfèrent aux documents originaux en allemand ou en anglais.

2.0. Recalculer avec l'extension `typearea.sty`

De nombreuses classes de LaTeX, y compris les classes standard, fournissent à l'utilisateur une division des espaces et de la zone de texte. Dans les classes standard de la distribution d'origine, les limites dépendent de la taille de la police choisie. En outre, il existe des packs comme géométrie (voir[Ume00]), donnant à l'utilisateur un contrôle total, avec la possibilité de paramétrer la zone de texte et les bordures.

KOMA-Script est livré avec le pack `typearea` qui fonctionne d'une manière quelque peu différente. Les paramètres et les automatismes proposés, basés sur un concept typographique, font qu'il est plus facile pour l'utilisateur de faire un bon choix.

Ce pack est expliqué dans la partie expert de ce manuel au chapitre 11. Cependant, les instructions consignées dans ce document sont plutôt destinées aux auteurs de classes et de packs.

2.1. Principes de base de la mise en page

En regardant une seule page d'un livre ou d'un ouvrage imprimé, on remarque, à première vue : un en-tête, un corps avec des bordures et un pied de page. On peut ajouter une espace entre la tête et le corps et entre le corps et le pied. Le corps est, dans le langage technique des typographes et le type de typographie une zone. La répartition de ces zones, ainsi que leur relation l'une à l'autre est décrite comme la mise en page ou composition. Je n'y connais pas grand chose en typographie, mais

ça ne me semble pas très technique comme définition.

Dans la littérature, plusieurs algorithmes et heuristiques sont proposés et discutés [Koh02] pour construire une bonne mise en page. Souvent, on trouve une méthode qui fonctionne avec des diagonales et leurs intersections. Le résultat recherché est que le rapport d'aspect du type de zone corresponde au rapport d'aspect de la page. Dans un document recto, les marges gauche et droite doivent avoir la même largeur, alors que les marges supérieure et inférieure doivent être dans le rapport de 1:2. Dans un document à double page, tel qu'un livre, il est bon de noter que la totalité de la largeur de la marge centrale doit être la même que celle de chacun des deux marges extérieurs. chacune des page contribue donc à la moitié de la marge centrale.

Dans le paragraphe précédent, la page mentionnée et soulignée est souvent prise à tort pour le format qui équivaldrait à la taille du papier. Cependant, quand on regarde un document relié, on peut voir qu'une partie du papier disparaît dans la reliure et que cette partie de la feuille n'est plus visible. Cependant, pour la zone, le format du papier n'a pas d'importance, mais plutôt celui de la page visible. Il est donc clair que la zone du type de pièce cachée par la reliure sur le format de papier doit être ajoutée à la bordure intérieure. Nous l'appelons correction de reliure, elle fait donc partie du calcul de la gouttière, sans bord intérieur visible.

La correction de reliure dépend du processus de production respectif et ne peut pas être déterminée en général. Il s'agit d'un paramètre redéfini pour chaque processus de production. Dans le domaine professionnel, cette valeur ne joue qu'un rôle mineur, car l'impression est généralement faite sur des feuilles de papier grand format qui seront réduites en conséquence. La coupe est faite de manière que les relations ci-dessus pour la page recto-verso soient maintenues.

Il n'y a marge de gauche et marge de droite que dans l'impression recto. En conséquence, bord interne et bord externe n'existent qu'en impression recto-verso. Dans les équations, nous travaillons avec une marge intérieure, parce que le bord interne est un élément de la double page complète.

Alors maintenant, nous connaissons les relations entre les différentes parties de la composition. Nous ne savons pas encore, ni sa largeur, ni sa hauteur. Nous pouvons choisir cependant une de ces dimensions et, avec le format du papier et la taille de la page, nous pouvons en déduire toutes les autres dimensions par la résolution de plusieurs équations mathématiques :

ratio hauteur : largeur de la zone de saisie=ratio hauteur : largeur de la page
hauteur de la marge supérieure : hauteur de la page-bord inférieur=1: 2
marge gauche: la marge de droite=1: 1 - demi bord interne: bord externe=1: 2

Largeur de page=largeur du papier-Reliure
bord supérieur de correction + marge inférieure=hauteur de la page - hauteur de zone
largeur marge gauche + marge de droite=largeur de la page - hauteur de zone
largeur marge intérieure + bord extérieur=largeur de la page -hauteur de zone
marge intérieure + correction de reliure=Gouttière

Bord : bord interne.

La question de la largeur de la zone de saisie est également discutée dans la littérature. La largeur de la zone de type optimale dépend de divers facteurs :

- Taille, largeur, type de la police utilisée
- Espacement des lignes
- Longueur des mots

L'influence de la police devient claire quand vous pensez à l'empatement. Il consiste en de petits traits horizontaux aux extrémités des barres de lettres. Les lettres finissant verticalement à la base de la ligne de texte ont tendance à entraîner le regard hors de cette ligne. Les empacements horizontaux aident à maintenir le regard aligné sur la ligne de base et facilitent le flot de lecture. Ils aident particulièrement à retrouver rapidement le début de la ligne après le saut depuis la fin de ligne précédente.

Ainsi une ligne de texte identique peut être plus longue dans le cas d'une police avec empatement que dans celui d'une police sans empatement.

L'espacement signifie la distance verticale entre les lignes de texte. Dans LaTeX l'espacement standard correspond à 20% de la taille de la police. Avec des commandes comme `\linespread` ou mieux à l'aide d'un pack comme `setspace` (voir [Tob00]), ces rapports peuvent être modifiés. Un espacement de ligne plus large permet à l'œil de mieux suivre cette ligne. Un écart très large facilite la lecture, mais perturbe l'œil qui doit parcourir de longues distances entre les lignes. En outre, un effet de bande très clair est perturbant pour le lecteur, désagréable à l'œil car la valeur de gris de la page est altérée dans ce cas. Malgré tout un espacement important permet une longueur de ligne plus importante.

Vous cherchez des valeurs concrètes pour des bonnes longueurs de ligne et avez peut être trouvé des informations différentes en fonction de l'auteur. Cela peut être est bien fondé dans la langue maternelle de l'auteur. L'œil habituellement saute de mot en mot, et les mots courts rendent cette tâche plus facile. Tous langages et polices confondus, une longueur de ligne de 60 à 70 caractères, espaces et ponctuation comprises, constitue un bon compromis. Cela suppose un espacement de ligne bien choisi. Avec les réglages par défaut de LaTeX vous n'avez pas de soucis à vous faire normalement. Une longueur plus importante est à réserver à l'usage de lecteurs aguerris, habitués à de longues heures quotidiennes de lecture.

Mais même dans ce cas des longueurs de ligne de 80 caractères ne sont pas raisonnables. Dans tous les cas, l'espacement doit être adaptée, 5 % à 10 % supplémentaires sont recommandés comme orientation. Dans le cas de certaines polices telles que Palatino, qui nécessitent pour une longueur de ligne normale déjà 5% d'espacement supplémentaire, des valeurs supérieures sont nécessaires.

Avant d'examiner la mise en page de la construction proprement dite, il y a encore des petites choses que vous devez savoir. Dans LaTeX la première ligne de la zone de texte d'une page ne commence pas au niveau du bord supérieur du bloc de texte. Le niveau de référence de la ligne est défini avec une distance minimale du bord supérieur de la zone de texte. Autres caractéristiques des commandes LaTeX `\raggedbottom` et `\flushbottom`. La première de ces commandes spécifie que la dernière ligne d'une page doit être positionnée là où cela a été calculé, ce qui peut provoquer une variation de la position verticale de la dernière ligne d'une page à l'autre qui peut atteindre jusqu'à une hauteur de ligne en cas de cumul d'ajustement de titres, de figures, de tableaux ou similaires, ce qui est généralement indésirable

dans les conditions d'impression double-face. La deuxième commande, `\flushbottom`, détermine que la dernière ligne est toujours au niveau du bord inférieur de la zone de texte. Pour atteindre ces compensations verticales, LaTeX peut avoir à s'étendre, ce qui est possible en additionnant les distances verticales extensibles. Une telle distance est, par exemple, l'espacement des paragraphes, mémorisée lors de l'utilisation de paragraphes sans espacements qui est la seule mesure verticale expansible, la hauteur de la zone de texte doit être un multiple de la hauteur de la ligne de texte, augmentée de la distance de la première ligne à partir du bord supérieur de la zone de texte.

En ce qui concerne les bases, les deux sections suivantes présentent les possibles constructions permises par KOMA-Script.

2.2. Mise en page par division

La manière la plus facile pour réaliser une zone de texte dans le même rapport que la page est la suivante:

- évaluer la partie BCOR nécessaire et diviser le reste de la page verticalement en rangées DIV d'égale hauteur.

- puis diviser la page horizontalement dans le même nombre de DIV en bandes de même largeur.

- prendre la ligne supérieure de la partie supérieure et les deux rangées du bas sur le bord inférieur. Pour une impression recto-verso, ne pas oublier la correction de reliure à l'intérieur et les deux colonnes sur les bords extérieurs.

- La marge intérieure est ensuite ajouté BCOR.

Il reste à déterminer la zone de texte dans la page. La largeur ou la hauteur des bordures fixe automatiquement le résultat de la zone de texte à partir du nombre de colonnes DIV. Comme les marges ont toujours besoin de trois lignes, DIV doit nécessairement être supérieur à trois. Le texte occupe horizontalement et verticalement au moins deux fois autant d'espace que les bords, DIV pourrait même être inférieur à 9. Avec cette valeur, la construction permet la mise en page classique dite au neuvième (voir la figure 2.1).

Sous KOMA-Script, la mise en page se réalise avec le pack de `typearea` pré-réglé pour le format papier A4. Selon la taille de la police, des valeurs différentes pour DIV peuvent être prises, voir le tableau 2.2. En l'absence de correction, on applique $BCOR=0$ pt et les résultats correspondent à peu près aux valeurs de mise en page du tableau 2.1.

En plus des valeurs prédéfinies, vous pouvez spécifier BCOR et DIV en chargeant le pack spécifié en option (voir section 2.6). En outre, une commande vous permet de calculer un ensemble explicitement et de transmettre ces valeurs comme paramètres existants. Le pack de `typearea` offre également, automatiquement, la possibilité de déterminer la valeur optimale de DIV qui dépendra de la taille de la police fixée au moment du recalcul.

Type de zone gauche

Type de zone droite

1	2	3	4	5	6	7	8	9		9	8	7	6	5	4	3	2	1
2																		2
3																		3
4																		4
5																		5
6																		6
7																		7
8																		8
9																		9

Mise en page gauche

Mise en page droite

Reliure de correction

fig 2.1 - double page de construction classique d'une grille avec application de DIV par neuf après déduction d'une correction de liaison

2.3. Mise en page en dessinant un cercle

En plus de la méthode de construction décrite précédemment, il existe dans la littérature une méthode un peu plus classique dite méthode médiévale ou du Moyen Age.

Dans cette méthode, vous trouverez les mêmes valeurs, non seulement dans la forme du rapport d'aspect, mais il est supposé, en outre, que l'optimum est obtenu lorsque la hauteur de la zone de texte est égale à la largeur de la page. La méthode exacte peut être lue dans [Tsc87].

Un inconvénient de ce procédé montre que la largeur de la zone de texte n'est plus dépendante de la police de caractères. Il est donc judicieux pour l'auteur ou le compositeur de sélectionner une bonne police pour cette zone de texte.

Avec typearea cette construction est modifiée dans la mesure où par la sélection d'une excellente-généralement absurde-valeur de DIV ou d'une option spéciale de pack, une valeur de DIV est déterminée avec laquelle la mise en page qui en résulte correspond aussi précisément que possible, aux valeurs parfaites de la page du livre médiéval. Il est donc encore une fois recours à la méthode de construction par division.

2.4. Option de sélection précoce ou tardive

Dans cette section, une particularité de KOMA-Script est présentée, qui concerne non seulement typearea mais aussi les autres packs et classes : l'utilisateur peut trouver toutes les informations sur un pack ou une classe, et cette section se retrouve, presque à l'identique, dans plusieurs chapitres. Les utilisateurs qui ne sont pas intéressés par ces instructions, mais qui souhaitent seulement avoir une vue globale de KOMA-Script, peuvent éviter cette étude plus approfondie du manuel.

```
\documentclass[Optionenliste]{KOMA-Script-classe}  
\usepackage[Optionenliste]{pack-Liste}
```

En LaTeX, il est prévu que les utilisateurs spécifient les options de classe sous la forme d'une simple liste de mots-clés séparés par des virgules comme un argument optionnel de `\documentclass`. Ces préférences sont passées à tous les packs qui comprennent l'option. De même, il est prévu que les utilisateurs spécifient les packs sous la forme d'une simple liste des mots-clés séparés par des virgules comme argument optionnel de `\usepackage`. KOMA-Script étend à d'autres possibilités le mécanisme d'options pour les classes KOMA-Script et les packs. En outre, la plupart des options de KOMA-Script ont une valeur. Une option n'est pas nécessairement de forme, mais peut aussi avoir l'option=valeur de forme. Sauf qu'il existe une différence de traitement data `\documentclass` et `\usepackage` KOMA-Script comme dans [Tea05b] ou toute introduction de LaTeX, par exemple [DGS + 12].

Lorsque vous utilisez une classe KOMA-Script, il n'est pas nécessaire d'en spécifier les options pour éviter une charge superflue. Cela est dû au fait que la classe charge, par défaut, les packs qui sont donc déjà installés.

Le réglage des options avec `\documentclass` ou `\usepackage` révèle un inconvénient majeur, les déclarations de longueurs, de compteurs et autres doivent être suivies strictement pour ne pas provoquer un message d'erreur. En effet, une extension de LaTeX dans la valeur d'une option est utilisée par ces instructions avant même que la valeur de KOMA-Script soit adoptée. L'affectation de valeurs avec des extensions ou compteurs de LaTeX ne devrait jamais se faire via `\usepackage\documentclass` mais toujours à l'aide des commandes `\KOMAOptions` ou `\KOMAOption`.

```
\KOMAOptions{liste d'options}  
\KOMAOption{option}{liste de valeurs}
```

KOMA-Script offre également la possibilité de changer la valeur des options avec la plupart des classes et des packs. Vous pouvez modifier les valeurs sélectivement avec la déclaration `\KOMAOptions`. Chaque option de la liste des options a la forme d'option=valeur.

Certaines options ont aussi une valeur par défaut (defaultValue). Si vous oubliez de préciser une valeur, c'est la valeur par défaut qui sera automatiquement appliquée. Certaines options peuvent avoir simultanément plusieurs valeurs. Pour ces options il est possible d'assigner à l'aide de `\KOMAOption` un ensemble de valeurs. Dans la liste des valeurs, les valeurs individuelles seront séparées les unes des autres par des virgules.

Si une valeur contient le signe égal ou une virgule, la valeur doit être incluse entre des accolades.

KOMA-Script utilise pour la réalisation de cette possibilité, les commandes `\FamilyOptions` et `\FamilyOption` de la famille "KOMA". Voir partie II, article 11.2,

Mettre une option avec une valeur non valide ou refusée produit une erreur. Si LaTeX est exécuté dans un mode interactif, il s'arrête à cet endroit. La saisie de "h" affiche un écran d'aide qui donne les valeurs valides pour l'option correspondante.

2.5. Compatibilité avec les versions précédentes de KOMA-Script

Réaliser ses documents dans le code source présente un plus grand intérêt lié au fait que les futurs développements de LaTeX devraient donner exactement le même résultat. Néanmoins, dans certains cas l'amélioration des packs ou la correction de bugs peuvent entraîner des modifications du comportement, en particulier pendant la transition vers la nouvelle version, ce qui n'est en général, ni souhaitable, ni voulu.

```
version=value  
version=first  
version=last
```

Depuis la version 2.96a de KOMA-Script, la version 2.9t de `scrlltr2` et la version 3.01b pour `typearea`, vous pouvez choisir si votre code source doit être intégré à la version courante de LaTeX ou figurer dans une nouvelle version et, si vous le souhaitez, participer à toutes les améliorations de cette nouvelle version. Vous pouvez choisir la version de KOMA-Script compatible avec la version de l'option, soit avec la version soutenue la plus ancienne, soit avec la version 2.9, 2, 9t ou une version ultérieure.

Si vous spécifiez une version inconnue comme valeur, un avertissement sera émis et pour des raisons de sécurité, `version=first` sera conservée.

La valeur `version=last` sélectionne la version la plus récente à chaque exécution de LaTeX, mais sachez que cette utilisation peut poser des problèmes de compatibilité avec l'évolution future de LaTeX. L'option sans valeur a la même fonction. C'est aussi la valeur par défaut tant que vous n'utilisez pas d'option obsolète. Dans ce cas, KOMA-Script passe automatiquement en `version=first` et un avertissement émis explique comment éviter cette commutation. Sinon, vous pouvez sélectionner un réglage différent compatible avec l'option obsolète.

La question de la compatibilité affecte principalement des questions de création. De nouvelles fonctionnalités qui ne sont pas liées à la transition, sont également disponibles si vous avez sélectionné la compatibilité d'option avec une version plus ancienne. Cette option n'a pas d'effet sur les modifications causées par l'élimination de bugs lorsque vous utilisez une version plus récente.

Cependant, si une compatibilité s'avère indispensable, pour cause de bug, il est préférable d'enregistrer et d'archiver la version de KOMA-Script utilisée avec votre document. Il est important de noter que la version après le chargement du pack `typearea` ne peut pas être modifiée. L'accès à `\KOMAOPTIONS` ou `\KOMAOPTION` n'est pas disponible.

2.6. Réglage de Typearea et mise en page

Le pack de `typearea` propose deux interfaces utilisateur différentes qui influent sur

la construction. Le réglage le plus important consiste à en spécifier les options qui peuvent ainsi être définies de différentes manières. Dans cette section, la classe « protokol » sera utilisée. Ce n'est pas une classe KOMA-Script, mais une classe hypothétique. Ce guide est basé sur le cas idéal, avec une classe correspondante disponible pour chaque tâche.

BCOR=Correction

Avec l'aide de l'option BCOR=correction vous entrez la valeur absolue de la correction de reliure, à savoir la largeur de la zone utilisée pour cette reliure. Cette valeur est prise en compte automatiquement dans la méthode de construction et ajoutée aux marges intérieure ou gauche, selon le montage choisi. Vous pouvez la corriger en lui spécifiant une unité de format reconnu par TEX.

Exemple :

Supposons que vous créez un rapport financier, imprimé au format A4, assemblé avec une reliure (clip wallet) de 7,5 mm. La taille du papier ne sera pas suffisante en ajoutant la reliure. Ainsi donc, il vous faudra écrire:

```
\documentclass[a4paper]{report}
\usepackage[BCOR=8.25mm]{typearea}
ou
\documentclass[a4paper, BCOR=8.25mm]{report}
\usepackage{typearea}
```

pour spécifier BCOR comme option globale.

Lorsque vous utilisez une classe de KOMA-Script, typearea est automatiquement chargé et il n'est pas nécessaire de l'appeler :

```
\documentclass[BCOR=8.25mm]{scrbook}
```

L'option a4paper peut être omise parce qu'elle est la valeur par défaut pour toutes les classes KOMA-Script.

Si vous choisissez une nouvelle valeur, celle qui est utilisée, par exemple,

```
\documentclass{scrbook}
\KOMAOptions{BCOR=8.25mm}
```

et si la classe est déjà en cours de chargement, ce sont les paramètres par défaut de scrbook qui sont utilisés. Lorsque vous changez de paramètres en utilisant l'une des commandes \KOMAOptions ou \KOMAoption, une nouvelle mise en page est calculée automatiquement en utilisant les nouveaux paramètres de marge.

Assurez-vous que, lorsque vous utilisez l'une des classes de KOMA-script, cette option est passée comme une option de classe soit par \KOMAOptions ou par \KOMAoption après le chargement de la classe, et que le pack de typearea est chargé et spécifié explicitement par \usepackage comme un argument optionnel lors de l'utilisation de KOMA-Script.

Les options \KOMAOptions ou \KOMAoption changées après le chargement du pack, entraînent automatiquement le recalcul de la mise en page et des marges.

DIV=facteur

L'option DIV=facteur détermine la méthode de construction : en combien de bandes la page est divisée horizontalement et verticalement. La méthode exacte peut être trouvée section 2.2. Il est important de savoir que n'importe quelle valeur entière peut être utilisée comme facteur à partir de 4, et que plus le facteur est grand, plus la zone de texte et les marges sont réduites. Il est bon de noter, cependant, que de très grandes valeurs peuvent mener à des conditions limites et nuire à une bonne méthode de construction, en fonction des autres options que vous aurez choisies. Ainsi, la tête peut être, dans certains cas extrêmes, en dehors de la page. Alors, utilisez l'option DIV=Facteur avec prudence, en conformité avec les principes de la ligne dite elle-même typographique.

Dans le tableau 2.1, la taille de la page au format A4 sans correction de certains facteurs de DIV détermine la taille des mises en pages. La suite dépend des contraintes de taille de police.

Exemple :

Supposons que pour écrire un procès-verbal de session, vous choisissiez d'utiliser la classe `protokol`, éditée en recto-verso, avec la police Bookman en 12 pt. Cette police est l'une des informations standard PostScript activée dans LaTeX avec la commande `\usepackage{bookman}`. Bookman est une police qui s'étale, c'est-à-dire que ses caractères individuels sont larges par rapport à sa hauteur. C'est pourquoi la valeur par défaut de DIV dans `typearea` est trop faible. Avec une valeur de 12 vous êtes au `tackage`. L'étude de ce chapitre, y compris les sections suivantes est basée sur DIV=15. Le protocole n'est pas lié et une correction de reliure n'est donc pas nécessaire.

Ainsi, vous écrivez:

```
\documentclass[a4paper, twoside]{protocole}
\usepackage{bookman}
\usepackage[DIV=15]{typearea}
```

Lorsque vous avez terminé, vous obtenez les journaux récents qui seront, à la fin du trimestre, tous liés ensemble comme un livre, aussi simplement qu'un collage. Pour cette liaison, y compris la reliure, un pli de 12 mm sera nécessaire, nécessitant le changement de l'option `typearea` en conséquence et l'utilisation de la classe pour les protocoles selon la norme ISO 9000:

```
\documentclass[a4paper, twoside]{iso9000p}
\usepackage{bookman}
\usepackage[DIV=15, BCOR=12mm]{typearea}
```

Bien sûr, vous pouvez aussi réutiliser une classe KOMA-Script:

```
\documentclass[twoside, DIV=15, BCOR=12mm]{scrartcl}
\usepackage{bookman}
```

L'option par défaut `A4paper` pourrait donc être supprimée.

Tableau 2.1 : Valeurs de mise en page en fonction de DIV pour A4 excluant \topskip

<i>DIV</i>	Typearea		Marges	
	Largeur	hauteur	haut	intérieur
6	105,00	148,50	49,50	35,00
7	120,00	169,71	42,43	30,00
8	131,25	185,63	37,13	26,25
9	140,00	198,00	33,00	23,33
10	147,00	207,90	29,70	21,00
11	152,73	216,00	27,00	19,09
12	157,50	222,75	24,75	17,50
13	161,54	228,46	22,85	16,15
14	165,00	233,36	21,21	15,00
15	168,00	237,60	19,80	14,00

(toutes les longueurs en mm)

Tableau 2.2 : Paramètres prédéfinis de DIV pour A4

Taille de la police de base:	10 pt	11 pt	12 pt
<i>DIV</i> :	8	10	12

Si vous utilisez KOMA-Script avec options de classe \KOMAOPTIONS ou \KOMAOPTION assurez-vous que DIV soit activé après le chargement de la classe et que typearea soit spécifié comme argument optionnel explicitement par \usepackage charge pour que les recalculs soient automatiques.

DIV=calc

DIV=classic

Comme mentionné dans la section 2.2, le format de papier A4 est un pré-réglage de valeur fixe pour DIV, présenté dans le tableau 2.2. qui à l'inconvénient de ne pas prendre en compte le suivi de la police utilisée. Cela peut se produire si la police A4 est plutôt étroite et rendre très rapidement désagréable la lecture d'un nombre trop élevé de lettres par ligne. Voir les considérations à la section 2.1.

Si un format de papier différent est sélectionné, le calcul typearea offrira une bonne valeur de DIV. Bien sûr, vous pouvez choisir ce calcul pour A4 et utiliser DIV=calc lieu de DIV=Factor. Bien sûr, vous pouvez également spécifier explicitement cette option pour tous les autres formats de papier. Agissez de même si vous souhaitez utiliser le calcul automatique car il est possible de remplacer les valeurs par défaut dans un fichier de configuration (voir la section 19.2).

La méthode de construction classique mentionnée dans la section 2.3 comme page d'un livre canonique médiéval est également sélectionnable. Dans ce cas, au lieu de DIV=Facteur ou simplement DIV=calc, utilisez DIV=classique. Il s'agit d'une valeur de DIV déterminée avec une bonne approximation possible de la page.

Exemple :

L'exemple des options répertoriées DIV=Facteur avec la police Bookman, pose exactement le problème d'une meilleure adéquation de la police en modifiant la valeur de DIV. Ainsi, dans le premier exemple déterminer cette valeur de typearea:

```
\documentclass[a4paper, twoside]{protocole}
\usepackage{bookman}
\usepackage[DIV=calc]{typearea}
```

Assurez-vous d'utiliser une des options de classe KOMA-Script ou paramétrez `\KOMAOptions` après le chargement de la classe. Si l'option via `\KOMAOptions` ou `\KOMAoption` est changée après le chargement du pack la mise en page et les bordures sont automatiquement recalculées avec `OptionDIV=classic`

```
DIV=current
DIV=last
```

Si vous avez suivi les exemples jusqu'à ici, vous savez comment obtenir le calcul du DIV en fonction de la police sélectionnée quand KOMA-Script-classe est utilisé avec un ensemble de polices.

Le problème pourrait venir de la classe KOMA-Script qui charge par défaut le pack `typearea` rendant impossible la saisie des options comme argument de `\usepackage`. Il n'est pas utile de préciser l'option `DIV=calc` comme argument de `\documentclass` qui serait directement activée lors du chargement du pack `typearea`. La mise en page et les marges seraient calculées pour la police LaTeX standard et ignorées pour le chargement de polices ultérieures.

Bien sûr, il est possible avec `\KOMAOptions{DIV=calc}` après chargement de calculer la mise en page et les bordures d'un nouveau type de pack de police ou avec `\KOMAoption{DIV}{calc}` où l'on précise ensuite une valeur de DIV et une valeur de calc. Cependant, comme il est souvent plus pratique de ne pas mettre en place l'option DIV après le chargement de police, mais dans un endroit bien en vue, par exemple charger une classe, `typearea` propose deux autres valeurs symboliques pour cette option.

Avec `DIV=current`, l'actualisation de la mise en page et des bordures d'impression est lancée, en utilisant exactement la valeur de DIV actuellement fixée, ce qui présente peu d'intérêt pour le recalcul de la mise en page si l'on choisit une police de base différente, mais qui, au contraire, est utile si vous voulez vous assurer de l'état limite du maintien de la DIV par le changement de l'excédent qui devrait représenter un multiple de `\baselineskip - \textheight - \topskip`.

Avec `DIV=last`, le recalcul de la mise en page et des bords impression est lancé, avec le même paramètre que celui utilisé dans le calcul précédent.

Exemple :

Supposons encore que pour la mise en page avec la police Bookman, une bonne longueur de ligne doit être calculée, et que, en même temps, on utilise une classe KOMA-Script. Rien de plus simple, on charge la valeur symbolique avec la commande `\KOMAOptions`:

```
\documentclass[BCOR=12mm, DIV=calc, twoside]{scrartcl}
\usepackage{bookman}
\KOMAOptions{DIV=last}
```

S'il est décidé d'utiliser ultérieurement une autre valeur de DIV, seul l'argument optionnel de `\documentclass` devra être changé.

Pour un résumé de toutes les valeurs symboliques possibles pour l'option DIV, voir le tableau 2.3. Il convient de noter à ce stade que l'utilisation du pack fontenc peut amener LaTeX à utiliser une police différente.

Tableau 2.3. Valeurs symboliques possibles pour la DIV option de déclaration ou DIV argument `\Typearea[BCOR]{DIV}`

areaset	Réorganiser la mise en page
calc	Recalculer la mise en page et déterminer la valeur de DIV pour ré-effectuer.
classic	Recalculer pour effectuer à nouveau après la page du livre canonique médiéval (calcul du cercle circonscrit).
current	recalcul à effectuer avec la valeur en cours de validité pour DIV.
default	Recalculer pour effectuer le passage à la valeur par défaut de la taille de la page en cours et de la police courante. Si aucune valeur par défaut n'existe, appliquer calc.
last	Recalcul avec le même DIV de l'argument donné à la dernière communication.

Le recalcul est souvent nécessaire dans le cadre de la variation de l'espacement des lignes étant donné que la mise en page doit être strictement calculée de telle sorte qu'un nombre entier de lignes entrent dans la zone de texte qui devra être elle-même recalculée ainsi que la zone de saisie normale pour l'interligne.

Par exemple, supposons que pour une thèse, la taille de police est de 10 pt avec une phrase demi-ligne nécessairement requise. LaTeX, normalement fixé à 10 pt, ajoute 2 pt, à savoir 1, 2 ligne. Il doit donc utiliser un facteur supplémentaire d'étirement de 1, 25. Supposons qu'une correction de reliure de 12 mm soit également nécessaire. La solution à ce problème pourrait être la suivante:

```
\documentclass[10pt, twoside, BCOR=12mm, DIV=calc]{scrreprt}
\linespread{1.25}
\KOMAOPTIONS{DIV=last}
```

Typearea exécute toujours la commande `\normalsize` pour calculer une nouvelle mise en page.

Le même exemple à l'aide du pack de `setspace` (voir[Tob00]) comme suit:

```
\documentclass[10pt, twoside, BCOR=12mm, DIV=calc]{scrreprt}
\usepackage[onehalfspacing]{setspace}
\KOMAOPTIONS{DIV=last}
```

Comme vous pouvez le voir dans l'exemple, utilisez le pack `setspace` qui, toutefois, ne s'applique qu'aux pt pour une police par défaut de tailles 10, 11 et 12 pt. Pour toutes les autres tailles de police, le pack utilise une approximation. A ce stade, il semble opportun de souligner que l'espacement des lignes de la page de titre doit être remis à la normale et les répertoires réglés avec l'interligne normale.

Exemple :

Un exemple complet serait:

```

\documentclass[10pt, twoside, BCOR=12mm, DIV=calc]
{scrreprt}
\usepackage[onehalfspacing]{setspace}
\AfterTOCHead{\singlespacing}
\KOMAOptions{DIV=last}
\begin{document}
\title{Titre}
\author{Papi Ray}
\begin{spacing}{1}
\maketitle
\end{spacing}
\tableofcontents
\chapter{Ok}
{document}

```

Voir également les notes dans la section 2.8. La commande `\AfterTOCHead` est présentée dans la partie II, chapitre 14.

Tableau 2.4. Arguments BCOR symboliques possibles `\typearea[BCOR]{DIV}`

current	Recalcule à nouveau avec la valeur BCOR en cours de validité.
---------	---

Rappeler vous que ces options sont disponibles pour une utilisation avec `\KOMAOptions` ou `\KOMAoption` après le chargement du pack, et qu'elles déclenchent un recalcul automatique de la mise en page du texte et des marges.

```

\typearea[BCOR]{DIV}
\recalctypearea

```

Si les options DIV ou BCOR accompagnent le pack `typearea`, la commande `\typearea` est appelée en interne. La valeur par défaut utilisée en interne lors de l'activation de l'option DIV pour BCOR, peut être trouvée dans le tableau 2.4 et si vous voulez recalculer la mise en page et les bordures de type pour DIV avec la valeur symbolique courante, vous pouvez utiliser directement `\typearea[current]{current}`

Si les valeurs BCOR et DIV sont modifiées, l'utilisation de `\typearea` est recommandée, car dans ce cas les bordures et les niveaux de réglage sont recalculés en une seule fois. D'autre part, `\KOMAOptions{DIV=DIV, BCOR=BCOR}` calcule les marges et établit de nouveaux niveaux pour DIV puis pour BCOR.

La commande `\typearea` est actuellement définie de sorte qu'il est également

possible de changer la mise en page d'un document en cours de création. Dans ce cas, cependant, les calculs se font sur une structure de base LaTeX qui change les valeurs internes et seuls, les changements nécessaires sont réalisés pour éliminer les risques d'erreurs.

Il est probable que ces fonctions figurent dans les futures versions de LaTeX 2ε, mais sans aucune garantie. L'utilisation dans un document nécessite toujours une pagination depuis `\KOMAOption{DIV}{last}`, `\KOMAoptions{DIV=last}` ou `\Typearea[current]{last}` qui sont nécessaires assez fréquemment pour le calcul de la mise en page et des bordures, ce qui peut être considéré comme un raccourci pour la commande `\recalctypearea`.

Exemple :

Si vous connaissez l'orthographe

```

\KOMAoptions{DIV=last}
ou
\typearea[current]{last}

```

Le recalcul de la mise en page et des bordures d'impression est trop lourd en raison des nombreux caractères spéciaux, vous pouvez simplement utiliser `\recalctypearea`

Tableau 2.5 : Valeurs standard pour de simples interrupteurs à KOMA-Script

valeur	action	valeur	action
true	Active l'option	false	Desactive l'option
on	Active l'option	off	Desactive l'option
yes	Active l'option	no	Desactive l'option

`twoside=valeur marche/arrêt`
`twoside=demi`

Comme expliqué dans la section 2.1, la mise en page nécessite de savoir si un document doit être placé sur un ou sur deux côtés. Dans un ensemble unilatéral, les marges gauche et droite sont de même largeur. En recto-verso, le bord intérieur d'une page est moitié moins grand que son bord extérieur respectif. Cette distinction, `twoside`, est une option `typearea` qui indique que le document est situé sur les deux côtés de la feuille.

Les valeurs marche-arrêt du tableau 2.5 peuvent être utilisées comme valeurs par défaut. Si une option est utilisée sans indication de valeur, c'est la valeur `true`, recto-verso, qui sera utilisée par défaut. Désactiver l'option et vous avez un document à une seule face.

Outre les valeurs du tableau 2.5 la valeur `demi` peut également être donnée. Cette valeur entraîne un double ensemble de bords unilatéraux, avec des marges non alternatives simple face, ou une composition double face avec des marges à une face, et des notes possibles en marge. Néanmoins, une correction de liaison est possible depuis KOMA-Script 3.12 (voir options BCOR) qui a ajouté une mise en page recto-verso sur le bord gauche des pages impaires et sur le bord droit des pages paires.

Mais si vous utilisez la compatibilité avec les versions antérieures de KOMA-Script (voir section 2.5), la correction de liaison fera partie de la marge de gauche sur les deux pages en utilisant `twoside=semi`.

L'option peut être utilisée comme option de classe dans `\documentclass`, comme option de package avec `\usepackage`, ou même après le chargement de l'ensemble de typearea avec `\KOMAOPTIONS` ou `\KOMAOPTION`. L'utilisation de l'option après le chargement automatique des packs de typearea permet le recalcul de la zone de type en utilisant `\recalctypearea`.

La composition recto-verso active avant la modification permet de créer un saut de page qui précédera la prochaine page impaire.

`twocolumn`=valeur marche/arrêt

Pour le calcul des types de zones à l'aide de `DIV=calc`, il est nécessaire de savoir si le document comporte une ou deux colonnes. Les considérations sur la longueur de ligne de la section 2.1 s'appliquent alors individuellement à chaque colonne, un texte mis en page sera deux fois plus large avec une seule colonne.

L'option `twocolumn` de typearea indique qu'une page du document comporte un texte aligné sur deux colonnes. Une valeur marche-arrêt par défaut peut être utilisée pour de simples interrupteurs du tableau 2.5. Si l'option est utilisée sans valeur spécifiée, la valeur `true` est supposée, qui indique le double colonnage. L'option désactivée indiquera, par défaut, une seule colonne.

L'option peut être réglée comme une option de classe `\documentclass`, comme une option de package dans `\usepackage` ou après chargement de type zone de texte `\KOMAOPTIONS` ou `\KOMAOPTION`. L'utilisation de cette option après le chargement typearea entraîne automatiquement le recalcul de la mise en page par `\recalctypearea`.

`headinclude`=valeur marche/arrêt

`footinclude`=valeur marche/arrêt

Jusqu'à présent, nous avons expliqué la construction de la mise en page en détaillant le contenu, la proportion entre bordures et contenu, mais une question cruciale reste posée : où se situe exactement la frontière entre les différents composants et que savons nous réellement de KOMA-Script?

À première vue, cette question paraît triviale : les frontières sont les parties de page, le pied, l'en-tête qui reste au-dessus, les marges gauche ou droite, le contenu. Mais ce n'est qu'une partie de l'histoire. Le bord extérieur n'est pas toujours vide. Dans certains cas, il pourra contenir des notes marginales (voir la commande `\exemple marginpar`, dans [DGS + 12] ou de l'article 3.21).

Pour le haut et le bas, l'important est de savoir de quelle façon sont traités l'en-tête et le pied de page. S'agit-il de deux entités gérées chacune d'une manière indépendante ou faut-il tenir compte du format et des bords respectifs ? Répondre à cette question n'est pas facile, mais il semble probable qu'un pied de page et un en-tête vides soient tangenciels et ne seront pas isolés lors de la composition de la page. L'en-tête ou le pied de page qui ne contient que la pagination, marquera une limite, sans impact sur l'effet visuel, sera identifié et reconnu comme tel lors de la lecture. Il devrait apparaître optiquement plus comme une marge et être inclus dans le présent. Il n'est pas pertinent, pour ce même effet visuel, de voir si le pied peut être facilement reconnu, lors de la lecture ou du survol, comme un pied ou pas. Ce qui importe est la façon dont une page bien remplie s'affiche avec la mise au point, lorsqu'on la regarde, ou comment elle peut entraîner une visualisation floue.

Une suggestion : demandez à vos grands-parents presbytes (en volant leurs lunettes) de lire la page à environ un demi mètre de la pointe de leur nez. En leur absence, essayez de lire la page tenue à bout de bras avec à vos propres yeux, mais en ne gardant qu'un seul œil ouvert. Les binoclars ont ici un net avantage : un pied de page qui contient non seulement le numéro, mais aussi des détails circonstanciels tels qu'un avis de droit d'auteur, ressemble plus, à distance, à une partie de corps de texte. Cet effet doit être pris en compte dans le calcul de la zone de saisie.

Pour l'en-tête, c'est un peu plus compliqué. Elle contient fréquemment le titre de la colonne 2. Dans le cas d'une création de rubriques avec des titres infinis de chapitres et de sections, les lignes d'en-tête seront très longues et sembleront faire partie du corps de texte. Cet effet devient encore plus significatif lorsque l'en-tête contient non seulement le chapitre ou le titre de section, mais aussi le numéro de page.

Avec du texte sur les côtés droit et gauche, la tête n'apparaîtra plus comme une marge vide, ce qui est moins évident avec le pied. La longueur des titres peut varier de sorte que la tête apparaisse dans la marge sur une page et le texte sur la page suivante. Pourtant, ces pages ne doivent, en aucun cas, être traitées différemment, ce qui conduirait l'en-tête à sauter verticalement. Dans ce cas, il est préférable d'inclure l'en-tête dans le cadre du texte.

La décision est facile lorsque le texte et l'en-tête ou le pied de page sont séparés par une ligne. Cela donne un aspect "fermé" et l'en-tête ou le pied de page semblent faire partie du corps de texte. Rappelez-vous: Il importe peu que la ligne améliore la séparation optique du texte et de l'en-tête ou du pied de page; seule l'apparence de la mise au point, lorsque l'on regarde, est importante.

Le pack de `typearea` ne peut pas prendre la décision de calculer séparément et de savoir s'il faut ou ne faut pas inclure en-têtes et pieds de page dans le cadre du corps de texte ou de la marge. Les options `headinclude` et `footinclude` permettent d'inclure l'en-tête ou le pied de page dans le cadre du texte. Ces options comprennent les valeurs standard pour de simples interrupteurs indiquées dans le tableau 2.5. On peut les utiliser sans spécifier une valeur. Dans ce cas, la valeur réelle est utilisée pour l'interrupteur simple. Pour mémoire, l'en-tête ou le pied de page sont comptés dans le cadre de la version anglaise du présent ouvrage.

Comme je l'ai écrit ci-dessus, améliorer grâce à une ligne de séparation qui isolerait l'en-tête ou le pied de page n'est pas pertinent ici. Le facteur décisif est l'observation floue par le `typearea` qui vous aide à prendre la décision d'intégrer au corps de texte, séparément, la tête et le pied en tant qu'options par les commandes `headinclude` et `footinclude`. Ces options comprennent en valeur unique, les standards pour les interrupteurs simples indiqués dans le tableau 2.5. Vous pouvez également l'utiliser sans options d'affectation et dans ce cas, imputer la valeur `true`, comme `marche-arrêt`, pour la tête ou le pied.

Si vous n'êtes pas sûr d'un réglage correct, rappelez vous que, le réglage par défaut est généralement `headinclude` et `footinclude=false`. Cela peut, cependant, en fonction de l'option de classe ou lors de l'utilisation d'autres packs changer avec les classes KOMA-Script (voir la section 3.1 et les chapitres 5 ou 6).

Vous devez vous rappeler que ces options doivent être précisées comme options de classe par `\KOMAOPTIONS` ou `\KOMAOPTION` après le chargement de l'une des

classes de KOMA-Script. Un changement après le chargement `typearea` ne conduit pas à un recalcul de la mise en page. Au contraire, le changement ne prend effet que lors du calcul de la nouvelle mise en page. Pour cette opération, voir l'option `DIV` avec les valeurs courantes ou la commande `\recalctypearea`.

Outre les documents dont la tête et le pied de la page appartiennent à la zone de texte qu'ils délimitent, on rencontre des documents qui affichent des notes marginales intégrées avec l'option `mpinclude`. Elles peuvent prendre une valeur standard comme indiqué dans le tableau 2.5. Vous pouvez également ne pas préciser l'option d'affectation et dans ce cas, `true` est utilisé comme valeur de marche-arrêt.

`mpinclude=`valeur marche/arrêt

L'effet de `mpinclude=true` est que la zone de texte est réduite d'une unité de largeur qui sera utilisée pour gérer l'espace réservé aux notes marginales. Avec `mpinclude=false`, qui est la valeur par défaut, une partie sera utilisée pour les notes de marge. Ce réglage, d'une unité en recto ou d'une moitié en recto-verso n'est pas recommandé et son usage restera réservé, en général, aux experts.

Dans la plupart des cas, l'option `mpinclude` est utile pour gérer un plus grand nombre de notes de marge, lorsqu'une partie de celles-ci doit être attribuée à la zone de texte. C'est par exemple le cas lorsqu'une ligne est utilisée pour une citation.

D'une manière habituelle, ces citations sont composées avec un positionnement du corps de texte déjà réglé, mais comme l'empreinte n'est pas fermée, les extrémités de flottement peuvent déborder partiellement dans la marge. On peut changer la possibilité d'utiliser `mpinclude` en augmentant la longueur `\marginparwidth` après le calcul de la zone de type ou encore en utilisant l'expression `\addtolength`. valeur pour élargir nécessitant une certaine finesse à l'usage. Voilà pourquoi l'utilisation de cette option est conseillée aux experts. Bien sûr, vous pouvez étendre la largeur de la marge, par exemple d'un tiers sur le bord, en procédant comme suit pour obtenir l'effet désiré:

```
\setlength{\marginparwidth}{1.5\marginparwidth}
```

Comme il n'existe actuellement aucune option pour fournir plus d'espace pour les notes de marge dans la zone de texte, il n'y a qu'une seule façon de le faire: adapter `\textwidth` et `\marginparwidth` après le calcul de la zone Type. Voir aussi `\AfterCalculatingTypearea` à l'article 19.1.

Vérifier que cette option est passée en utilisant l'une des classes KOMA-Script comme une option de classe ou par `\KOMAOPTIONS` ou `\KOMAOPTION` après le chargement de la classe. Changer cette option après le chargement `typearea` ne conduit pas à un calcul automatique, il ne concerne que la prochaine mise en page. Pour un nouveau calcul, charger l'option `DIV` avec les valeurs courantes ou charger la commande `\recalctypearea`.

`headlines`=nombre de lignes
`headheight`=Hauteur

Nous avons vu comment configurer une mise en page en utilisant le pack de

typearea et comment spécifier si en-tête et pied de page font partie du texte ou des marges. Pourtant, il manque encore quelques précisions, par exemple l'indication de la hauteur de l'en-tête utilisée pour les titres et le nombre de lignes. Ces résultats sont obtenus avec les options headlines et headheight.

L'option headlines indique le nombre de lignes de l'en-tête. En principe, typearea fonctionne avec la valeur 1,25 par défaut qui est un compromis assez grand pour fournir l'espace à un en-tête souligné (voir section 3.1) et assez petit pour ne pas grignoter la bordure lors de l'utilisation d'un simple travail avec un en-tête non souligné. Ainsi, la valeur de la manchette par défaut conviendra dans la plupart des mises en page mais vous devrez l'adapter dans certains cas particuliers.

Exemple :

Supposons l'en-tête avec un texte comportant deux lignes, ce qui signifie, sur chaque côté, qu'un avertissement de trop plein `\vbox "` sera délivré par LaTeX.

Pour l'éviter typearea calculera une mise en page appropriée :

```
\documentclass[a4paper]{article}
\usepackage[headlines=2.1]{typearea}
```

Il est possible et même recommandé lors de l'utilisation d'une classe KOMA-Script de passer cette option directement à la classe:

```
\documentclass[headlines=2.1]{scrartcl}
```

Les commandes qui définissent le contenu des deux lignes de l'en-tête peuvent être trouvées dans les chapitres 5 ou 6.

Dans certains cas, il est utile de spécifier directement en tant que valeur la longueur et la hauteur de la tête, ce qui est possible à l'aide de l'option alternative headheight dont LaTeX connaît, par défaut toutes les longueurs et les tailles possibles. Cependant, pour mieux maîtriser votre présentation, il est préférable de paramétrer une option telle que `\baselineskip` au moment du calcul de la mise en page et des bordures. Rappelez vous que ces options doivent être données comme options de classe soit par `\KOMAOPTIONS` ou par `\KOMAOPTION` après le chargement de classe en utilisant l'une des classes KOMA-Script.

Un changement de ces options après le chargement typearea ne conduit pas à un calcul automatique mais affecte plutôt le changement jusqu'au nouveau calcul de la prochaine mise en page. Pour connaître les modalités de recalcul de la mise en page, voir l'option DIV avec les valeurs courantes ou la commande `\recalctypearea`.

footlines=nombre de lignes
footheight=hauteur

De même que pour la tête, la hauteur du pied n'est jamais indiquée, il faut donc en préciser les options. Par conséquent typearea définit pour la mise en place, une nouvelle hauteur et une nouvelle longueur du pied si elles n'existent pas. Elles seront ensuite utilisées, par exemple, dans les classes et les packs pour la conception de l'en-tête et du pied en fonction des classes et des packs utilisés. Le faisceau KOMA-Script `scrlayer-scrpage` prend en compte la hauteur du pied `\footheight`, contrairement à `scrpage2`, et travaille donc activement avec

typearea. Néanmoins, les classes KOMA-Script ne reconnaissent pas `\footheight` qui, sans le pack `assistance`, ne fournit que des styles de page avec une seule ligne dans le pied de page.

Comparable à celle de `headlines`, l'option de hauteur de `footlines` est le rapport du pied de page, et normalement `typearea` fonctionne avec 1, 25. Cette valeur est un compromis assez grand pour fournir l'espace pour un pied surligné ou souligné (voir section 3.1), et d'autre part assez petit pour ne pas empiéter sur la bordure quand on opère avec un pied de page simple sans ligne de séparation. Ainsi, la valeur par défaut dans la plupart des cas types est une bonne valeur, mais il vous faudra parfois régler pour des besoins ponctuels, comme pour l'en-tête, la hauteur du pied avec plus de précision.

Exemple :

Supposons que, dans le pied, une déclaration de droits d'auteur doit être réglé sur deux lignes. Bien que cette fonction soit présente par défaut dans LaTeX, ce dernier ne teste pas l'espace affecté pour les pieds, ce qui peut entraîner un déséquilibre dans la répartition de la mise en page et des bordures. L'ensemble `scrlayer - scrpage` qui gère le contenu pourrait être pré-réglé après examen minutieux de cette même mise en page, ce qui rendra nécessaire, lors du calcul de la mise en page, de préciser une plus grande hauteur de pied :

```
\documentclass[a4paper]{article}
\usepackage[footlines=2.1]{typearea}
```

Il est également possible de passer cette option directement en utilisant une classe KOMA-Script :

```
\documentclass[footlines=2.1]{scrartcl}
```

Les commandes définissant le contenu d'un pied de page de deux lignes peuvent être trouvées dans le chapitre 6.

Dans certains cas, il est utile de spécifier directement en tant que valeur de la longueur, la hauteur de la base, ce qui est possible à l'aide de l'option `\footheight`. Il convient de noter cependant, que l'utilisation d'une longueur de LaTeX comme `\baselineskip` demande un réglage de l'option indispensable au moment du calcul de la mise en page et des bordures.

Comme d'habitude, rappelez vous que ces options doivent être données comme options de classe par `\KOMAOPTIONS` ou `\KOMAOPTION` après chargement d'une classe en utilisant KOMA-Script. Un changement de ces options après le chargement de `typearea` ne mène pas à un calcul automatique de la mise en page, le changement ne concerne que le calcul de la prochaine mise en page, voir la commande `\recalctypearea` ou l'option `DIV` avec les valeurs courantes.

```
\areaset[BCOR]{largeur}{hauteur}
```

Jusqu'à présent, beaucoup a été dit ici sur la façon de créer une bonne mise en page pour application standard, que `typearea` peut rendre plus facile à l'utilisateur, mais en même temps l'influencer. Cependant, il y a aussi des cas où le corps du texte sera exactement maintenu à une certaine taille, sans s'occuper de la conception de la mise en page ni de contraintes supplémentaires, les bordures réparties, autant que possible, pour le mieux et le cas échéant avec la prise en

compte d'une reliure de correction.

L'option `typearea` nous offre `\areaset` pour cette fonction, avec les paramètres largeur et hauteur de la zone de texte à côté de la correction de reliure. Les bordures sont ensuite calculées automatiquement en prenant en compte les paramètres en-tête et pied de page des options `\headinclude` et `\footinclude`, mais les options titre et hauteur de la tête sont ignorées dans ce cas !

Exemple :

Supposons un texte, de type article imprimé sur du papier A4 avec une largeur exacte de 60 caractères en utilisant la police Typewriter et 30 lignes par page. Vous pouvez travailler avec le préambule suivant :

```
\documentclass[a4paper, 11pt]{article}
\usepackage{typearea}
\newlength{\CharsLX}% Largeur de 60 caractères
\newlength{\LinesXXX}% Hauteur de 30 lignes
\settowidth{\CharsLX}{\texttt{1234567890}}
\setlength{\CharsLX}{6\CharsLX}
\setlength{\LinesXXX}{\topskip}
\addtolength{\LinesXXX}{29\baselineskip}
\areaset{\CharsLX}{\LinesXXX}
```

Le facteur de 29 au lieu de 30 est justifié par le fait que la base de la première ligne est déjà sur le bord supérieur de la mise en page aussi longtemps que la hauteur de la rangée du haut est inférieure à `\topskip`. La ligne supérieure ne nécessite aucune hauteur, la ligne inférieure du projet marque la limite de la mise en page.

Si au contraire vous voulez un livre de poésie avec une zone carrée de texte de longueur de côté de 15 cm et une marge de reliure de 1 cm, ce qui suit est conseillé:

```
\documentclass{poesie}
\usepackage{typearea}
\areaset[1cm]{15cm}{15cm}

\areaset[actuel]{\textwidth}{\textheight}
```

DIV=`areaset`

Dans de rares cas, il peut être utile de réaligner la mise en page actuellement sélectionnée en utilisant l'option `DIV=areaset`,

```
\KOMAOptions {DIV=areaset}{déclaration}
\areaset[current]{\textwidth}{\textheight}
```

Le même résultat est également obtenu avec `DIV=last`. La mise en page est alors définie par le dernier `areaset`.

Le pack `typearea` n'a pas été fait pour gérer des valeurs de marge prédéfinies. Si vous devez le faire, vous pouvez utiliser le pack géométrie (voir [Ume00])

2.7. Choix et réglage de la taille du papier

Le format de papier est une caractéristique de base indispensable à la création d'un document, comme la notion d'ensemble de la construction et de sa mise en page (voir les sections 2.1 à 2.3) avec le choix de la taille de papier. Alors que les classes LaTeX standard sont limitées à quelques formats anglo-saxons, KOMA-Script a été créé pour gérer, avec le pack KOMA-Script typearea, les formats européens, et même des pages avec des tailles inhabituelles.

paper=Format

L'option de format du papier est l'élément central de sélection dans KOMA-Script, inspiré initialement par le format légal d'une lettre américaine. En outre, le format ISO des lignes A, B, C et D est possible, ainsi par exemple, A4 ou a4. Les formats sont pris en charge avec en option la valeur indiquant un format figure ou un format paysage. Ici, les paysages ne diffèrent des figures que par les programmes dvips qui s'articulent autour de -90° pour le paysage tout en tournant à 90° pour une figure. Le format figure est surtout utile quand un programme post-script affiche des pages à l'envers en mode paysage. Ainsi, la différence joue un rôle et même l'option pagesize est une possibilité qui doit être utilisée. En outre, le format peut également prendre la largeur de la forme avec hauteur à préciser. Il convient de noter que jusqu'à la version 3, 01a la hauteur et la largeur étaient inversées, ce qui devrait être pris en compte en particulier lors de l'utilisation d'un paramètre de compatibilité approprié (voir la version de l'option, section 2.5).

Exemple :

Supposons que vous vouliez une carte d'index au format ISO A8 imprimée paysage, sans pied de page ni en-tête avec des bordures à la cote minimale.

```
\documentclass{article}
\usepackage[headinclude=false, footinclude=false, %
            paper=A8, paper=landscape]{typearea}
\areaset{7cm}{5cm}
\pagestyle{empty}
\begin{document}
\section*{format du papier}
letter, legal, executive, a0, a1 \dots\%
b0, b1 \dots\c0, c1 \dots\d0, d1 \dots
\end{document}
```

Il existe des cartes au format spécial (largeur: 5 cm hauteur: 3 cm) qui rendent possible l'encodage suivant:

```
\documentclass{article}
\usepackage[headinclude=false, footinclude=false,
            paper=5cm:3cm]{typearea}
\areaset{4cm}{2.4cm}
\pagestyle{empty}
\begin{document}
\section*{Definierte papierformate}
letter, legal, executive, a0, a1 \dots\%
b0, b1 \dots\c0, c1 \dots\d0, d1 \dots
\end{document}
```

Dans les classes standard LaTeX, le format utilisé par défaut est américain lettre. KOMA-Script travaille avec du papier A4 orienté portrait.

Cette option doit être transmise en utilisant l'une des classes KOMA-Script comme option de classe avec `\KOMAOPTIONS` ou `\KOMAOPTION`. Un changement de l'orientation ou de la taille du papier après le chargement `typearea` ne conduit pas à un calcul automatique de la zone de type, il ne concerne que le nouveau calcul de la prochaine zone de saisie. A ce propos, voir l'option `DIV` avec les valeurs courantes ou la commande `\recalctypearea`.

`pagesize=opérateur de sortie`

Les descriptions ci-dessus permettent de sélectionner le format du papier selon les normes adaptées à notre système européen. Elles ont seulement une influence sur la sortie, mais elles redéfinissent et remplacent les paramètres internes de LaTeX. La fonction `typearea` l'utilisera ensuite dans la distribution de la page, pour positionner les marges et la zone de texte.

La spécification DVI format peut être faite à tout moment avant le formatage du papier. Elle permet une sortie directe au format DVI vers un langage d'imprimante de bas niveau comme `PLC3*` ou `ESC/P24`, ce qui a peu d'importance, généralement, mais fixe le point 0 de référence DVI en haut à gauche. Traduit en langage PostScript ou PDF, ce point est ailleurs et la taille du papier doit être spécifiée dans le fichier de sortie, information manquante.

Pour résoudre ce problème, les pilotes respectifs utilisent un format de papier prédéfini que l'utilisateur peut changer soit au moyen d'une option ou par spécification dans le fichier source TEX. Lorsque vous utilisez le pilote DVI `dvips`, l'information ne peut être donnée que sous la forme d'une commande spéciale avec `pdfTEX` ou `VTEX` pour mettre en place deux dimensions.

L'option `pagesize` permet de régler le format du papier, pour lequel un pilote gère le document de sortie. Les pilotes de sortie pris en charge peuvent être trouvés dans le tableau 2.6. La valeur par défaut est `pagesize=false`. L'utilisation de l'option sous la forme `pagesize` sans valeur correspondant à la valeur `pagesize=auto`.

Exemple :

Supposons que vous vouliez un document utilisé comme un fichier DVI, qui permette également de créer une version en ligne au format PDF. Le préambule pourrait par exemple commencer comme ceci :

```
\documentclass{article}
\usepackage[paper=A4, pagesize]{typearea}
```

`pdfTEX` est maintenant utilisé pour le traitement et la sortie PDF est activée. Les deux dimensions spéciales `\pdfpagewidth` et `\pdfpageheight` sont fixées en conséquence. Toutefois, un fichier DVI peut être généré si LaTeX ou `pdfLaTeX` est lancé par une commande spéciale écrite au début de ce fichier.

Il est recommandé de toujours spécifier l'option `pagesize`. En règle générale, la méthode sans pilote de sortie est idéale.

Tableau 2.6: pilotes de sortie pour l'option `pagesize=output`

auto

Si les registres pdfTeX spécifiques `\pdfpagewidth` et `\pdfpageheight` sont présents, le pilote de sortie activé est `pdftex` et le pilote utilisé est `dvips`.

automedia

Gestion des pilotes de la fonction de sortie automatique. Gère et fixe en plus les registres des médias spécifiques VTeX, `\mediawidth` et `\mediaheight` largeur et hauteur du support.

false, no, off

Le format de papier n'est pas signalé au pilote de sortie.

dvipdfmx

Le format de papier est décrit `\special {pageSize=largeur, hauteur}` dans le fichier DVI. Le nom du pilote de sortie provient du fait que le programme `dvipdfmx` inclut un commutateur de format de papier.

dvips

Lorsqu'il est utilisé dans le préambule du document, le format du papier spécial est, par défaut, celui utilisé dans le document `\{pageSize=largeur, hauteur}`. Depuis `dvips`, il n'y a plus de commutateur de taille ce qui nécessite un hack plutôt compliqué pour réaliser la modification, à utiliser avec précaution, à vos propres risques!

pdftex

Le format du papier est sur les registres spécifiques pdfTeX `\pdfpagewidth` et dans la série `\pdfpageheight`, ce qui est également possible à tout moment dans le document.

2.8. Conseils

En particulier, pour la présentation d'un travail écrit pendant ses études, rapport ou thèse, il existe de nombreuses dispositions régulièrement contournées telle la typographie établie contre toutes les règles et qui en viole les principes les plus élémentaires. La raison en est bien souvent l'incompétence typographique de ceux qui les publient. Elle peut se justifier à l'origine par l'utilisation d'une machine à écrire mécanique ou par l'usage d'un traitement de texte basique datant du début des années 1980, avec lesquels, l'un ou l'autre, il n'était pas possible de produire une sortie typographiquement correcte sans effort considérable, mais la typographie idéale existe-t-elle ?

Ainsi, des règles ont été créées qui semblent réalisables sans peine et permettent une correction facile. Celles-ci comprennent alors des paramètres avancés utiles pour l'impression recto avec une machine à écrire. Pour éviter les interlignes courts composés par étourderie, les marges ont été maintenues étroites et l'espacement des lignes augmenté à 1,5 pour les corrections. Avant l'avènement des systèmes modernes de traitement de texte, une simple interligne aurait été la seule alternative à TEX.

Dans un tel document, même des signes de correction auraient été difficile à ajouter. Lorsque les ordinateurs sont devenus plus largement disponibles pour le

traitement de texte, certains étudiants ont utilisé de "belles" polices, en pensant améliorer l'aspect visuel du résultat de leur travail. Il est facile d'oublier cependant que ces polices sont plus difficiles à lire et donc impropres à ce nouvel usage

Ainsi, deux types de corps de lettres devenus largement utilisés semblaient bien adaptés et pourtant... prenons le cas de Times police relativement étroite conçue au début du 20ème siècle pour les colonnes des journaux britanniques et que l'on retrouve dans les versions modernes, fréquemment prescrites mais ne correspondant pas vraiment à une spécification de taille.

Le développement de LaTeX rend son usage de plus en plus fréquent, avec une possibilité accrue pour les corrections efficaces de lignes sensibles.

Pourtant, il est souvent difficile de mettre en œuvre des règles typographiques discutables ou délicates lorsque seul, un nombre réduit en sera respecté lors de l'utilisation d'une police non-proportionnelle, dont il existe cependant quelques bons modèles. Un texte dans une écriture qui semble bonne peut parfois être amélioré, par exemple, avec une base en saillie, petit "i" ou "l" donnant une autre largeur de caractère, utilisation d'un empattement justifié typographiquement car compatible avec l'écriture habituelle de notre langue française. Certains paramètres peuvent être ignorés lors de l'utilisation de LaTeX ou interprétés généreusement comme "utiliser 60 caractères par ligne" non pas comme valeur fixe, mais comme valeur moyenne ou maximale.

Les paramètres d'enregistrement sont généralement destinés à obtenir lors de leur exécution un résultat utile, même si l'auteur ne sait pas ce qui doit être observé. Utile signifie : lisible et corrigé et à mon avis, un ensemble LaTeX avec l'option `typearea` pour la mise en page répond, dès le début, à ces exigences. Si vous êtes confronté à des règles sensiblement différentes, je vous suggère de tester la mise en forme de votre texte avant de fournir la version définitive de votre travail. En cas de besoin, ce dernier pourrait être ajusté en modifiant la mise en page à l'aide de l'option `DIV\areaset` que l'auteur vous déconseille mais que je vous recommande de n'utiliser qu'avec précaution.

Dans le pire des cas, on peut faire usage du pack géométrie (voir [Ume00]), qui ne fait pas partie de KOMA-Script, ou modifier les paramètres de mise en page dans LaTeX lui-même. On peut trouver les valeurs déterminées par `typearea` dans le fichier journal de votre document, ce qui rend possible des ajustements modérés. Toutefois vous devez absolument veiller à ce que les proportions de la zone de texte correspondent approximativement à celles de la page en cours de correction avec sa reliure.

S'il s'avérait absolument nécessaire de définir l'espacement de texte d'une demi-ligne, ne définissez aucun `\baselinestretch`. Cette approche trop souvent recommandée, est obsolète depuis l'introduction de LaTeX 2 ϵ en 1994. Dans le pire des cas, utilisez l'instruction `\linespread` mais je vous recommande le pack `setspace` bien qu'il ne fasse pas partie de KOMA-Script (voir [Tob00]). Vous devrez recalculer la mise en page de `typearea` après la conversion de l'interligne et revenir à l'espacement de lignes normal pour le titre, pour le contenu des tables et des listes diverses.

En outre, le pack `setspace` propose ses propres instructions pour la gestion des environnements spécifiques de la bibliographie et de l'index.

Le pack `typearea` adapte à l'aide de l'option `DIV=calc` une zone de texte généreuse

et de nombreux typographes traditionalistes indiquent que la longueur de la ligne qui en résulte est encore trop grande.

La valeur DIV calculée figure dans le fichier journal pour chaque document et vous pouvez la modifier sans peine après la première exécution de LaTeX.

Il n'est pas rare que l'on me pose cette question : pourquoi, sur ce projet de mise en page, écrire chapitre par chapitre qui semblent rabâcher lors de recalculs, alors qu'il serait plus facile d'écrire un seul ouvrage qui permettrait de définir les marges comme un traitement de texte.

J'ai souvent pensé qu'un tel ensemble serait sans doute la meilleure solution, et tout le monde sait que, pour choisir la bonne marge, les bordures de KOMA-Script ne sont pas la formule idéale. Permettez-moi pourtant de conclure cette section par une citation de Hans Peter Friedrich Willenberg et de Forssmann, deux maîtres en typographie des plus distingués. voir[WF00])

« Faire soi-même est depuis longtemps l'habitude, une longue habitude, les résultats sont souvent discutables car les simples typographes ne voient pas ce qui est faux et ne peuvent pas savoir ce qui est important. Alors, vous vous habituez à une typographie douteuse, parfois mauvaise. [...] Et pourtant, une génération pourrait venir qui reprendrait goût à la chose typographique, avec l'envie de l'appliquer à la décoration ... »