L'extension fancyhdr
Mise en page en LaTeX

Piet van Oostrum *
Département d'informatique
Université d'Utrecht

16 août 2016

Résumé

Cet article décrit comment personnaliser la mise en page de vos documents LaTeX, autrement dit comment changer les marges et tailles des pages, les entêtes et pieds de page et comment placer convenablement les tables et figures (collectivement appelés flottants) sur une page.

À l'origine, cet article constituait la documentation de l'extension fancyheadings. Il contenait également d'autres informations comme l'utilisation avancée des marques. Il fait maintenant l'objet d'évolutions pour contenir d'éléments comme la gestion des flottants. La documentation de fancyheadings a été mise à jour pour se conformer à la version 2 de cette extension [1]. Pour des raisons de compatibilité avec certains systèmes d'exploitation, le nom de l'extension est devenu fancyhdr.

Bien que ce document utilise des commandes LaTeX 2_ε, la plupart des techniques présentées peuvent être utilisées avec des versions plus anciennes de LaTeX en faisant les changements adéquats.

Table des matières

*Une part considérable de cet article a été écrit par George Grätzer (Université du Manitoba) dans *Notices Amer. Math. Soc.* Merci, George !

1. Cette version est due à Real Soon Now.

1 Introduction

Une page dans un document LaTeX est construite à partir de divers éléments comme montré en figure 1. Le corps contient le texte principal du document ainsi que les « flottants » (tables et figures).

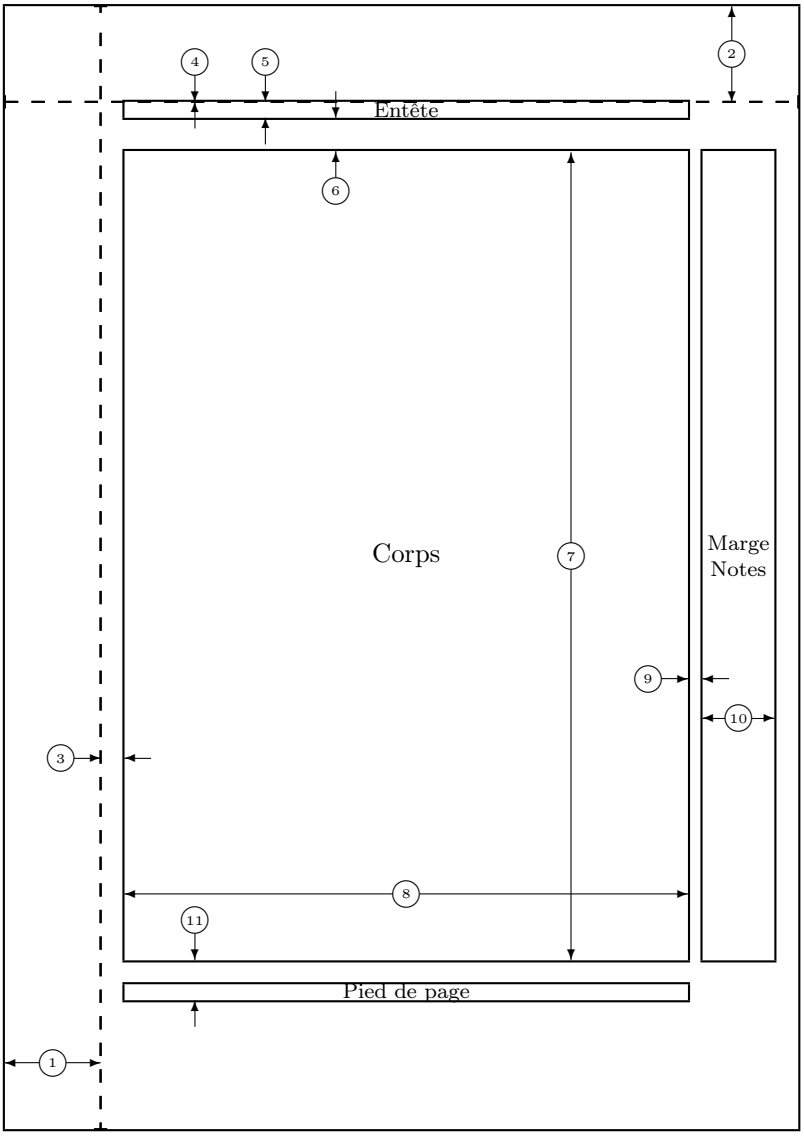
Les pages sont construites par la routine de sortie de LaTeX que, du fait de sa relative complexité, nous ne chercherons pas à modifier. Certaines des extensions décrites dans cet article intégrent de légères modifications de la routine de sortie pour accomplir des choses impossibles autrement. Vous devriez utiliser ces extensions pour obtenir le résultat désiré plutôt que de bidouiller avec la routine de sortie vous-même.

Il y a un certain nombre de choses dont vous devez être bien conscient :

1. les marges à gauche ne sont pas appelées `\leftmargin`[2] mais `\evensidemargin` (pour les pages paires) et `\oddsidemargin` (pour les pages impaires). Dans les documents en recto uniquement, il s'agit de `\oddsidemargin`. `\leftmargin` est également un paramètre de LaTeX valide mais a un usage différent (en l'occurrence, l'indentation des listes).
2. La plupart des paramètres ne devraient pas être changés au cours du document. Certains changements peuvent fonctionner lors d'un saut de page. Si vous souhaitez changer la taille d'une seule page, vous pouvez utiliser la commande `\enlargethispage`.

Les zones de notes marginales contiennent de petits éléments d'information créés par la commande `\marginpar`. Dans les documents recto-verso, les notes marginales apparaissent alternativement à gauche et à droite. Les notes marginales ne sont pas à des places fixes dépendant du papier mais approximativement à la même hauteur que les paragraphes dans lesquelles elles apparaissent. Du fait de l'algorithme retenu pour décider du placement des notes marginales, elles peuvent

2. N.D.T. : *leftmargin* signifie « marge de gauche », tandis que *evensidemargin* signifie « marge de côté pair » et *evensidemargin* « marge de côté impair ».



1	un pouce + \hoffset	2	un pouce + \voffset
3	\oddsidemargin = 18pt	4	\topmargin = 0pt
5	\headheight = 12pt	6	\headsep = 25pt
7	\textheight = 609pt	8	\textwidth = 424pt
9	\marginparsep = 11pt	10	\marginparwidth = 54pt
11	\footskip = 30pt		\marginparpush = 5pt (non affiché)
	\hoffset = 0pt		\voffset = 0pt
	\paperwidth = 597pt		\paperheight = 845pt

FIGURE 1: Éléments d'une page. Les valeurs montrées sont celles qui s'appliquent à ce document, pas les valeurs par défaut.

apparaître dans un document recto-verso du mauvais côté si elles sont proches d'un saut de page. Si vous souhaitez placer de l'information à des endroits fixes dans les marges, vous pouvez vous servir de la technique décrite dans les sections 20 et 21.

La première partie de cet article décrit comment changer les zones d'entête et de pied de page. La seconde partie décrit comment placer vos flottants à l'endroit désiré.

2 Entêtes et pieds de page

Les entêtes et pieds de page dans LaTeX sont définis par les commandes `\pagestyle` et `\pagenumbering`. La première définit le contenu général des entêtes et pieds de page (par exemple où le numéro de page est imprimé) tandis que la seconde définit le format du numéro de page. LaTeX a quatre styles de page standards :

`empty`	pas d'entête et de pied de page
`plain`	pas d'entête et un pied de page contenant le numéro de page centré
`headings`	pas de pied de page et un entête contenant le titre du chapitre ou de la section ou de la sous-section et le numéro de page
`myheadings`	pas de pied de page, un entête contenant le numéro de page et de l'information fournie par l'utilisateur

Bien que ces styles soient utiles, ils sont assez limités. Des styles de page complémentaires peuvent être définis par des commandes de la forme `\ps@xxx`. Ce document est exécuté quand une commande `\pagestyle{xxx}` est mise dans le document. La commande `\ps@xxx` doit définir les commandes suivantes associées au contenu de l'entête et du pied de page :

`\@oddhead`	entête des pages impaires dans un document recto-verso (sur toutes les pages pour un document en recto uniquement)
`\@evenhead`	entête des pages paires dans un document recto-verso
`\@oddfoot`	pied de page des pages impaires dans un document recto-verso (sur toutes les pages pour un document en recto uniquement)
`\@evenfoot`	pied de page des pages impaires dans un document recto-verso

Ce ne sont pas des commandes utilisateurs mais plutôt des « variables » qui sont utilisées par la routine de sortie de LaTeX. Les noms de commandes contenant le caractère « `@` », ils doivent être définis dans un fichier d'extension ou être placés entre les commandes `\makeatletter` et `\makeatother`.

La commande `\pagenumbering` définit la mise en forme du numéro de page. Elle a un paramètre tiré de la liste suivante :

`arabic`	chiffres arabes
`roman`	chiffres romains minuscules
`Roman`	chiffres romains majuscules
`alph`	lettres minuscules
`Alph`	lettres majuscules

La commande `\pagenumbering{xxx}` définit la commande `\thepage` comme le développement du numéro de page dans le format indiqué avec `xxx`. La commande `\pagestyle` intègre ensuite la commande `\thepage` à la place souhaitée. Par ailleurs, `\pagenumbering` réinitialise le numéro de page à 1. Les commandes `\pagestyle` et `\pagenumbering` s'appliquent à la page qui est en cours de construction : elles doivent donc se trouver à un endroit où la page qu'elles vont impacter est clairement définie (voir section 18).

3 Qu'est donc fancyhdr ?

L'extension fancyhdr vous permet de personnaliser vos entêtes et pieds de page dans LaTeX de façon simple. Vous pouvez définir des entêtes et pieds de page :

— en trois parties ;
— avec des filets décoratifs ;
— plus larges que le corps du texte ;
— sur plusieurs lignes ;
— distincts entre pages paires et pages impaires ;
— particuliers pour les pages des débuts de chapitre ;
— particuliers pour les pages avec des flottants.

Bien sûr, vous avez le contrôle intégral sur les fontes, affichages en majuscules et minuscules, et ainsi de suite.

4 Utilisation simple de fancyhdr

Pour utiliser cette extension dans un document LaTeX 2_ε, le fichier `fancyhdr.sty` doit être placé dans un répertoire où TeX peut le trouver (par exemple dans le même répertoire que le fichier à compiler), et inclure dans le préambule du document après

```
\documentclass{...}
```

les commandes [3] :

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

Nous pouvons visualiser la mise en page que nous pouvons obtenir avec fancyhdr de la façon suivante :

Entête de gauche	Entête du centre	Entête de droite
	corps de la page	
Pied de page de gauche	Pied de page du centre	Pied de page de droite

Les entêtes et pieds de page de gauche sont justifiés à gauche ; les entêtes et pieds de page du centre sont centrées ; les entêtes et pieds de droite sont justifiés à droite.

Nous définissons chacun de ces six « champs » et les deux filets décoratifs séparément.

5 Un exemple simple

K. Grant écrit un rapport à Dean A. Smith, sur « La performance des nouveaux diplômés » avec la mise en page suivante :

3. Dans le cas de LaTeX 2.09, vous devez indiquer `[fancyhdr]` dans la commande `\documentstyle` au lieu de la commande `\usepackage`.

où « 3 » est le numéro de la page. Le titre, « La performance des nouveaux diplômés » est mis en gras.

Ceci s'obtient avec les commandes suivantes à la suite de `\pagestyle{fancy}`[4] (la commande `\thepage` affiche le numéro de page courant. `\bfseries` est une des commandes permettant de sélectionner en LaTeX 2ε une ✖ police ✖ grasse) :

```
\lhead{}
\chead{}
\rhead{\bfseries La performance des nouveaux diplômés}
\lfoot{De: K. Grant}
\cfoot{À: Dean A. Smith}
\rfoot{\thepage}
\renewcommand{\headrulewidth}{0.4pt}
\renewcommand{\footrulewidth}{0.4pt}
```

C'est maintenant bien, à ceci près que la première page n'a pas besoin d'entête et pied de page. Pour éliminer ces éléments à l'exception du numéro de page centré près, il faut saisir

```
\thispagestyle{plain}
```

après les commandes `\begin{document}` et `\maketitle`.

Vous pouvez également saisir

```
\thispagestyle{empty}
```

si vous ne voulez ni entête, ni pied de page.

En fait, dans les classes standards LaTeX, la commande `\maketitle` est définie de telle sorte qu'une commande `\thispagestyle{plain}` y est automatiquement placée. Aussi, si vous *voulez* avoir une présentation personnalisée sur une page contenant `\maketitle`, vous devez saisir une commande `\thispagestyle{fancy}` après `\maketitle`.

6 Un exemple d'impression recto-verso

Certaines classes de document, telle `book.cls`, propose une mise en page recto-verso par défaut : les pages paires et impaires ont des présentations différentes, d'autres classes utilisent l'option `twoside` pour proposer de la mise en forme recto-verso.

Essayons maintenant de faire notre rapport en recto-verso. La présentation vue plus haut sera utilisée pour les pages impaires (pages de droite) et la présentation suivante servira pour les pages paires (pages de gauche) :

4. Notez que la version 1 de fancyheadings utilisait la commande `\setlength` pour changer les paramètres `\...rulewidth`.

La performance des nouveaux diplômés		
	corps de la page	
4	De : K. Grant	À : Dean A. Smith

où « 4 » est le numéro de page.

Voici les commandes :

```
\fancyhead{} % vide tous les entêtes
\fancyhead[RO,LE]{\bfseries La performance des nouveaux diplômés}
\fancyfoot{} % vide tous les pieds de page
\fancyfoot[LE,RO]{\thepage}
\fancyfoot[LO,CE]{De : K. Grant}
\fancyfoot[CO,RE]{À : Dean A. Smith}
\renewcommand{\headrulewidth}{0.4pt}
\renewcommand{\footrulewidth}{0.4pt}
```

Nous avons utilisé les commandes plus générales `\fancyhead` et `\fancyfoot`. Elles ont des paramètres additionnels entre crochets qui permettent de préciser pour quelles pages et/ou partie de l'entête ou du pieds de page elles s'appliquent. La première commande `\fancyhead` n'a pas ce paramètre et s'applique donc à l'ensemble des champs d'entête. En général, ce n'est utile que pour se débarasser de tous les réglages par défaut ou définitions précédentes, comme dans le cas présent. De la même manière, la première commande `\fancyhead` sans paramètre s'applique sur tous les champs de pied de page et les vide donc ici tous. Dans le cas présent, cela aurait pu ne pas être fait car l'ensemble des champs de pied de page sont redéfinis juste après.

Les sélecteurs qui peuvent être utilisés entre les crochets sont indiqués dans la figure 2. Ils peuvent être combinés comme dans `\fancyhead[LE,RO]{texte}` qui définit alors le champ d'entête de gauche pour les pages paires et le champ d'entête de droite pour les pages impaires. Si vous n'indiquez pas un E ou un O, les définitions s'appliquent par défaut sur toutes les pages. De même pour les sélecteurs L, R et C. Vous pouvez noter par ailleurs que `\lhead` dans la précédente section est une juste une abréviation de `\fancyhead[L]`. Les sélecteurs peuvent être saisis en majuscules comme en minuscules.

Il existe aussi une commande plus générale, `\fancyhf`, que vous pouvez utiliser pour combiner les spécifications des entêtes et des pieds de page. Ceci demande d'utiliser les nouveaux sélecteurs H (entête) et F (pied de page). En fait, `\fancyhead` et `\fancyfoot` ne sont que `\fancyhf` avec H et F présaisis.

Une nouvelle fois, vous pouvez utiliser `\thispagestyle{plain}` pour une mise en page simple de la page 1.

7 Redéfinition du style `plain`

Certaines commandes LaTeX, comme `\chapter`, utilisent la commande `\thispagestyle` pour basculer automatiquement en style de page `plain`, ignorant ainsi les effets du style de page alors en cours. Pour personnaliser également de telles pages, vous devez redéfinir le style de page `plain`. Comme indiqué précédemment, vous pourriez le faire en travaillant sur la commande `\ps@plain`,

E	page paire (*even*)
O	page impaire (*odd*)
L	champ gauche (*left*)
C	champ du centre (*center*)
R	champ droit (*right*)
H	entête (*header*)
F	pied de page (*footer*)

FIGURE 2: Sélecteurs

mais fancyhdr vous donne une façon plus simple de le faire avec la commande `\fancypagestyle`. Cette comande peut être utilisée pour redéfinir tout syle de page existant (comme `plain`) ou en définir de nouveaux, si par exemple une partie de votre document doit utiliser un autre style de page. Cette commande a deux paramètres : le premier est le nom du style de page à définir, le second une série de commandes changeant l'entête et le pied de page, autrement dit `fancyhead` et autres. Sont aussi autorisées les commandes changeant `\headrulewidth` et `\footrulewidth`. Par exemple, redéfinissons le style `plain` pour le rapport de la section 6 en mettant en gras le numéro de page.

```
\fancypagestyle{plain}{%
\fancyhf{} % vide tous les entêtes et pieds de page
\fancyfoot[C]{\bfseries \thepage} % sauf le centre
\renewcommand{\headrulewidth}{0pt}
\renewcommand{\footrulewidth}{0pt}}
```

8 La mise en page par défaut

✖Let us use the `book.cls` documentclass and the default settings for fancyhdr ; so we only issue the commands

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

and let fancyhdr take care of everything. On the pages where new chapters start, we get a centered page number in the footer ; there is no header, and there are no decorative lines.

On an even page, we get the layout :

1.2 EVALUATION *CHAPTER 1. INTRODUCTION*

page body

4

On an odd page, we get the layout :

page body

where the header text is slanted uppercase.

This default layout is produced by the following commands :

```
\fancyhead[LE,RO]{\slshape \rightmark}
\fancyhead[LO,RE]{\slshape \leftmark}
\fancyfoot[C]{\thepage}
```

The following settings are used for the decorative lines :

`\headrulewidth`	0.4pt
`\footrulewidth`	0 pt

The header text is turned into all uppercase in `book.cls`.

9 The scoop on LaTeX's marks

Usually, for documents of class `book` and `report`, you may want to use chapter and section information in the headings (chapter only for one-sided printing), and for documents of class `article`, section and subsection information (section only for one-sided printing). LaTeX uses a marker mechanism to remember the chapter and section (section and subsection) information for a page ; this is discussed in detail in the LaTeX *Companion*, Section 4.3.1.

There are two ways you can use and change the higher- and lower-level sectioning information available to you. The macros : `\leftmark` (higher-level) and `\rightmark` (lower-level) contain the information processed by LaTeX, and you can use them directly as shown in Section 8.

The `\leftmark` contains the Left argument of the *Last* `\markboth` on the page, the `\rightmark` contains the Right argument of the *fiRst* `\markboth` or the only argument of the *fiRst* `\markright` on the page. If no marks are present on a page they are "inherited" from the previous page.

You can influence how chapter, section, and subsection information (only two of them !) is displayed by redefining the `\chaptermark`, `\sectionmark`, and `\subsectionmark` commands [5]. You must put the redefinition after the first call of `\pagestyle{fancy}` as this sets up the defaults.

Let us illustrate this with chapter info. It is made up of three parts :

— the number (say, 2), displayed by the macro `\thechapter`
— the name (in English, Chapter), displayed by the macro `\chaptername`
— the title, contained in the argument of `\chaptermark`.

Figure 3 shows some variants for "Chapter 2. Do it now" (the last example is appropriate in some non-English languages). The % signs at the end of the lines are to prevent unwanted space. Normally you would continue the lines and remove these % signs [6].

5. There are similar commands for `paragraph` and `subparagraph` but they are seldom used.

6. the `\MakeUppercase` command is used in LaTeX 2_ε to generate uppercase text, while in LaTeX 2.09 `\uppercase` is used. The difference is that `\MakeUppercase` also deals with non-ASCII letters. Fancyhdr defines `\MakeUppercase` to be an alias for `\uppercase` if it isn't defined.

Code :	Prints :

`\renewcommand{\chaptermark}[1]{%`
`\markboth{\chaptername`
`\ \thechapter.\ #1}{}}`
	Chapter 2. Do it now

`\renewcommand{\chaptermark}[1]{%`
`\markboth{\MakeUppercase{%`
`\chaptername}\ \thechapter.%`
`\ #1}{}}`
	CHAPTER 2. Do it now

`\renewcommand{\chaptermark}[1]{%`
`\markboth{\MakeUppercase{%`
`\chaptername\ \thechapter.%`
`\ #1}}{}}`
	CHAPTER 2. DO IT NOW

`\renewcommand{\chaptermark}[1]{%` `\markboth{#1}{}}`	Do it now
`\renewcommand{\chaptermark}[1]{%` `\markboth{\thechapter.\ #1}{}}`	2. Do it now

`\renewcommand{\chaptermark}[1]{%`
`\markboth{\thechapter.%`
`\ \chaptername.\ #1}{}}`
	2. Chapter. Do it now

FIGURE 3: Marker variants

For the lower-level sectioning information, do the same with `\markright`.

So if "Section 2.2. First steps" is the current section, then

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

will give "2.2. First steps"

Redefining the `\chaptermark` and `\sectionmark` commands may not eliminate all uppercaseness. E.g. the bibliography will have a title of BIBLIOGRAPHY in the header, as the `\MakeUppercase` is explicitly given in the definition of `\thebibliography`. Similar for INDEX etc. If you don't want to redefine these commands, you can use the `\nouppercase` command that fancyhdr makes available in the header and footer fields. Note that this may screw other things, like uppercase roman numerals in your headers, so it should be used with care. Essentially this command typesets its argument in an environment where `\MakeUppercase` and `\uppercase` are changed into do-nothing operations.

```
\lhead{\nouppercase{\rightmark}}
\rhead{\nouppercase{\leftmark}}
```

It should be noted that the LaTeX marking mechanism works fine with chapters (which always start on a new page) and sections (which are reasonably long). It does not work quite as well with short sections and subsections. This is a problem with LaTeX, not with fancyhdr.

As an example let's take a page layout where the leftmarks are generated by the sections and the

rightmarks by the subsections (as is default in the `article` class). Take a page with some short sections, e.g.

Section 1.
subsection 1.1
subsection 1.2
Section 2.

As the leftmark contains the *last* mark of the page it will be "Section 2.", and the rightmark will be "subsection 1.1" as it will be the *first* mark of the page. So the page header info will combine section 2 with subsection 1.1 which isn't very nice. The best you can do in these cases is use only the `\rightmark`s and redefine `\sectionmark` accordingly. A LaTeX command `\firstleftmark` would also be a nice addition (see the `extramarks` package in section 19).

Another problem with the marks in the standard LaTeX classes is that the higher level sectioning commands (e.g. `\chapter`) call `\markboth` with an empty right argument. This means that on the first page of a chapter (or a section in article style) the `\rightmark` will be empty. If this is a problem you must manually insert extra `\markright` commands or redefine the `\chaptermark` (`\sectionmark`) commands to issue a `\markboth` command with two decent parameters.

As a final remark you should also note that the `*` forms of the `\chapter` etc. commands do *not* call the mark commands. So if you want your preface to set the header info but not be numbered nor be put in the table of contents, you must issue the `\markboth` command yourself, e.g.

```
\chapter*{Preface\markboth{Preface}{}}
```

Entering the `\markboth` command inside the `\chapter*` insures that the mark will not be separated from the title by a page break. Of course with `\chapter*` this wouldn't be a problem if you put the mark command after the chapter title, as the `\chapter*` command starts at a new page. However with a `\section*` it could be dangerous to say :

```
\section*{Preface}
\markboth{Preface}{}
```

as a page break may occur between the two commands.

10 Dictionary style headers

Dictionaries and concordances usually have a header containing the first word defined on the page or both the first and the last words. This can easily be accomplished with fancyhdr and LaTeX's `mark` mechanism. Of course if you use the marks for dictionary style headers, you cannot use them for chapter and section information, so if there are also chapters and sections present, you must redefine the `\chaptermark` and `\sectionmark` to make them harmless :

```
\renewcommand{\chaptermark}[1]{}
\renewcommand{\sectionmark}[1]{}
```

Now you do a `\markboth{#1}{#1}` for each dictionary or concordance entry `#1` and use `\rightmark` for the first entry defined on the page and `\leftmark` for the last one.

If you want to use a header entry of the form firstword–lastword it would be nice if this would be reduced to just the form firstword if both are the same. This could happen if there is just one

entry on the page. In this case a test must be made to check if the marks are the same. However, TEX's marks are strange beasts, which cannot be compared out of the box with the plain TEX `\if` commands. Fortunately the `ifthen` package works well :

```
\newcommand{\mymarks}{
  \ifthenelse{\equal{\leftmark}{\rightmark}}
    {\rightmark} % if equal
    {\rightmark--\leftmark}} % if not equal
\fancyhead[LE,RO]{\mymarks}
\fancyhead[LO,RE]{\thepage}
```

Dictionaries are often done with two columns. Unfortunately there is a bug in LATEX's twocolumn option which causes some marks to be lost. If you use David Carlisle's fix2col.sty this will be solved.

11 Fancy layouts

You can make a multi-line field with the `\\` command. It is also possible to put extra space in a field with the `\vspace` command. Note that if you do this you will probably have to increase the height of the header (`\headheight`) and/or of the footer (`\footskip`), otherwise you may get error messages "Overfull `\vbox` ... has occurred while `\output` is active"[7]. See Section 4.1 of the LATEX *Companion* for detail.

For instance, the following code will place the section title and the subsection title of an article in two lines in the upper right hand corner :

```
\documentclass{article}
\usepackage{fancyhdr}
\pagestyle{fancy}
\addtolength{\headheight}{\baselineskip}
\renewcommand{\sectionmark}[1]{\markboth{#1}{}}
\renewcommand{\subsectionmark}[1]{\markright{#1}}
\rhead{\leftmark\\\rightmark}
```

You can also customize the decorative lines. You can make the decorative line in the header quite thick with

```
\renewcommand{\headrulewidth}{0.6pt}
```

or you can make the decorative line in the footer disappear with

```
\renewcommand{\footrulewidth}{0pt}
```

The decorative lines, themselves, are defined in the two macros `\headrule` and `\footrule`. For instance, if you want a dotted line rather than a solid line in the header, redefine the command `\headrule` :

```
\renewcommand{\headrule}{\vbox to 0pt{\hbox
    to\headwidth{\dotfill}\vss}}
```

7. If you use `11pt` or `12pt` you will probably also have to do this, because LATEX's defaults are quite small

There is one additional parameter that you can set : `\footruleskip`. It defines the distance between the decorative line in the footer and the top of the footer text line. By default it is set to 30% of the normal line distance. You may want to adjust it if you use unusally large or small fonts in the footer. Change it with `\renewcommand`.

12 The width of the headers and footers

The width of headers and footers is `\headwidth`, which by default equals the width of the text : `\textwidth`. You can make the width wider (or narrower) by using the commands `\fancyheadoffset`, `\fancyfootoffset` and `\fancyhfoffset` [8]. These are similar to `\fancyhead`, `\fancyfoot` and `\fancyhf`. The only difference is that the offsets don't accept the C option.

Usage : `\fancyhfoffset[place]{length}` and similar for the others.

It defines offsets to be applied to the header/footer to let it stick into the margins if length > 0 and to be smaller if length < 0. place is like in `\fancyhead`, except that only E,O,L,R (or the lowercase) can be used.

With these commands `\headwidth` will be dynamically calculated in the headers/footers to be `\textwidth` + left offset + right offset, so that e.g. you can use the following to define a red line in the header and a blue line in the footer (please note that the width of the footer is also called `\headwidth` although it may differ from the width of the header) :

```
\renewcommand{\headrule}{{\color{red}%
  \hrule width\headwidth height\headrulewidth \vskip-\headrulewidth}}

\setlength{\footrulewidth}{\headrulewidth}

\renewcommand{\footrule}{{\color{blue}%
  \vskip-\footruleskip\vskip-\footrulewidth
\hrule width\headwidth height\footrulewidth\vskip\footruleskip}}
```

13 Two book examples

The following definitions give an approximation of the style used in Leslie Lamport's LaTeX book.

Lamport's header overhangs the outside margin. This is done with the offset commands described in the previous section.

```
\documentclass{book}
\usepackage{fancyhdr}
\pagestyle{fancy}
\usepackage{calc}
\fancyheadoffset[LE,RO]{\marginparsep+\marginparwidth}
\renewcommand{\chaptermark}[1]{\markboth{#1}{}}
\renewcommand{\sectionmark}[1]{\markright{\thesection\ #1}}
\fancyhf{}
```

8. These commands are defined since fancyhdr version 3.0. In older versions you just changed `\headwidth` with the `\setlength` and `\addtolength` commands. You are encouraged to use these newer offset commands instead. You certainly can't mix the two approaches.

```
\fancyhead[LE,RO]{\bfseries\thepage}
\fancyhead[LO]{\bfseries\rightmark}
\fancyhead[RE]{\bfseries\leftmark}
\fancypagestyle{plain}{%
   \fancyhead{} % get rid of headers
   \renewcommand{\headrulewidth}{0pt} % and the line
}
```

Notice that the `\chaptermark` and `\sectionmark` commands have been redefined to eliminate the chapter numbers and the uppercaseness.

For the second example, we take the $\mathcal{AMS}$-LaTeX book.

Chapter pages have no headers or footers. So we declare

```
\thispagestyle{empty}
```

for every chapter page, and we do not need to redefine plain.

Chapter and section titles appear in the form : 2. DO IT NOW, so we have to redefine `\chaptermark` and `\sectionmark` as follows (see Section 9) :

```
\renewcommand{\chaptermark}[1]%
   {\markboth{\MakeUppercase{\thechapter.\ #1}}{}}
\renewcommand{\sectionmark}[1]%
   {\markright{\MakeUppercase{\thesection.\ #1}}}
```

In an even-header, the page number is printed as the LeftHeader and the chapter info as the RightHeader ; in an odd-header, the section info is printed as the LeftHeader and the page number as the RightHeader. The CenteredHeaders are empty. There are no footers.

There is a decorative line in the header. It is 0.5pt wide, so we need the commands :

```
\renewcommand{\headrulewidth}{0.5pt}
\renewcommand{\footrulewidth}{0pt}
```

The font used in the headers is 9 pt bold Helvetica. The PSNFSS system by Sebastian Rahtz uses the short (Karl Berry) name `phv` for Helvetica, so this font is selected with the commands :

```
\fontfamily{phv}\fontseries{b}\fontsize{9}{11}\selectfont
```

(See Sections 7.6.1 and 11.9.1 of the LaTeX *Companion*.) Let us define a shorthand for this :

```
\newcommand{\helv}{%
   \fontfamily{phv}\fontseries{b}\fontsize{9}{11}\selectfont}
```

Now we are ready for the page layout :

```
\documentclass{book}
\usepackage{fancyhdr}
\pagestyle{fancy}
\renewcommand{\chaptermark}[1]%
```

```
   {\markboth{\MakeUppercase{\thechapter.\ #1}}{}}
\renewcommand{\sectionmark}[1]%
   {\markright{\MakeUppercase{\thesection.\ #1}}}
\renewcommand{\headrulewidth}{0.5pt}
\renewcommand{\footrulewidth}{0pt}
\newcommand{\helv}{%
   \fontfamily{phv}\fontseries{b}\fontsize{9}{11}\selectfont}
\fancyhf{}
\fancyhead[LE,RO]{\helv \thepage}
\fancyhead[LO]{\helv \rightmark}
\fancyhead[RE]{\helv \leftmark}
```

14 Special page layout for float pages

Some people want to have a special layout for float pages (pages only containing floats). As these pages are generated autonomically by LaTeX, the user doesn't have any control over them. There is no `\thispagestyle` for float pages and any change of the page style will at least also affect the page before the float page. With fancyhdr, however, you can specify in each of the header- or footer fields

`\iffloatpage{`value for float page`}{`value for other pages`}`

You can even use this to get rid of the decorative line on float pages only by defining :

```
\renewcommand{\headrulewidth}{\iffloatpage{0pt}{0.4pt}}
```

Sometimes you may want to change the layout also for pages that contain a float on the top of the page or a float on the bottom of the page.

fancyhdr gives you the commands `\iftopfloat` and `\ifbotfloat` similar to `\iffloatpage`.

Note : Marks in floats will not be visible in LaTeX's output routine, so it is not useful to put marks in floats. So there is currently no way to let a float (e.g. a figure caption) influence the page header or footer.

15 Those blank pages

In the book class when the `openany` option is not given or in the `report` class when the `openright` option is given, chapters start at odd-numbered pages, half of the time causing a blank page to be inserted. Some people prefer this page to be completely empty, i.e. without headers and footers. This cannot be done with `\thispagestyle` as this command would have to be issued on the *previous* page. There is, however, no magic necessary to get this done :

```
\clearpage{\pagestyle{empty}\cleardoublepage}
```

As the `\pagestyle{empty}` is enclosed in a group it only affects the page that may be generated by the `\cleardoublepage`. You can of course put the above in a private command. If you want to have this done automatically at each chapter start or when you want some other text on the page then you must redefine the `\cleardoublepage` command.

```
\makeatletter
\def\cleardoublepage{\clearpage\if@twoside \ifodd\c@page\else
  \hbox{}
  \vspace*{\fill}
  \begin{center}
    This page intentionally contains only this sentence.
  \end{center}
  \vspace{\fill}
  \thispagestyle{empty}
  \newpage
  \if@twocolumn\hbox{}\newpage\fi\fi\fi}
\makeatother
```

16 N of M style page numbers

Some document writers prefer the pages to be numbered as n of m where m is the number of pages in the document. There is a package `nofm.sty` available, but some versions of it are defective, and most don't work with fancyhdr because they take over the complete page layout. For LaTeX 2$_\varepsilon$ there is a package `lastpage` available which you can use with fancyhdr as follows :

```
\usepackage{lastpage}
...
\cfoot{\thepage\ of \pageref{LastPage}}
```

If you are still using LaTeX2.09 and you are not able to switch to LaTeX 2$_\varepsilon$ you can use the LaTeX2.09 compatible `lastpage209.sty`. which is defined as follows :

```
\let\origenddocument=\enddocument
\def\enddocument{\clearpage\if@filesw
  {\addtocounter{page}{-1} \immediate\write\@mainaux
    {\string\newlabel{LastPage}{{}{\thepage}}}}\origenddocument}
```

The value of the `LastPage` label can be used to make different headers or footers on the last page of a document. E.g. if you want the footer of every odd page, except if it is the last one, to contain the text "please turn over", this can be done as follows [9] :

```
\usepackage{lastpage}
\usepackage{ifthen}
...
\rfoot{\ifthenelse{\isodd{\value{page}} \and \not
    \value{page}=\pageref{LastPage}{please turn over}{}}
```

17 Chapter or section related page numbers

In technical documentation very often page numbers are used of the form 2-10 where the first number is the chapter number and the second is the pagenumber relative to the chapter. Sometimes

9. This requires a reasonably recent version of the `ifthen` package.

section is used rather than chapter. The package chappg can be used to get this format. If you want to change the layout e.g. use a dot rather than a dash, or a section rather than a chapter, you should make a private copy and edit it.

This package redefines `\thepage` as `\arabic{chapter}-\arabic{page}`. Unfortunately this gives numbers rather than letters for appendices. A better definition would be `\thechapter-\arabic{page}` but you can give this definition yourself after the `\usepackage{chappg}` command. What the package also does is reset the page number to 1 at the beginning of each chapter.

There is a fundamental difference between the page numbering of the style "m of n" as described in the previous section and the current one. The m of n style is only used in the page header or footer, but not in the table of contents, index, or references like "*See page* xx". Therefore it does not change the command `\thepage`. The page numbering style "2-10", however should be used in all references to the page number, therefore it must be done by redefining `\thepage`.

18 When to change the headers and footers ?

Sometimes you want to change the header or footer layout in the course of a document. Some of these changes can be accomplished by using the mark mechanism as may be seen in section 9 and 19. However, sometimes we want a more drastic change, e.g to change the page numbering from roman to arabic (with `\pagenumbering`), to change one of the fancyhdr fields or to change to another page style. Sometimes you may be surprised to find the change to occur too early. In general the above mentioned changes take effect immediately, i.e. on the page that is currently being built. If you want the change to take effect at the next page you must make sure that the current page is finished. In most cases this can be done by issuing a `\clearpage` command before any of the above mentioned changes. If this is not possible you can use the `afterpage` package with :
`\afterpage{\lhead{new value}}` or `\afterpage{\pagenumbering{roman}`. You cannot use `\afterpage` to change the `\pagestyle` as the commands issued by `\afterpage` are local in a group, and the `\pagestyle` command makes only local changes. The `\pagenumbering` and the fancyhdr commands make global changes so they will work, as will the `\thispagestyle` command.

It should be noted that although the fancyhdr commands like `\fancyhead` take effect immediately, this does not mean that any "variables" used in these commands get the value they have at the place where these commands are given. E.g. if `\fancyfoot[C]{\thepage}` is given the page number that will be inserted in the footer is not the page number of the page where this command is given, but rather the page number of the actual page where the footer is constructed. Of course for the page number this is what you expect, but it is also true for other commands.

So if you have a book where each chapter is written by a different author and you want the name of the author in the lower left-hand corner you can use the following commands :

```
\newcommand{\TheAuthor}{}
\newcommand{\Author}[1]{\renewcommand{\TheAuthor}{#1}}
\lfoot{\TheAuthor}
```

and start each chapter with the command `\Author{Real Name}`. If however, the author name would be changed before a page is completed the wrong author could come in the footer. This would be the case if you gave the above command *before* the `\chapter` command rather than after it. Another source of problems is the fact that TEX's output routine processes commands ahead,

so it may already have processed some commands that produce text that will appear on the next page. See the next section for an example.

19 Headers and footers induced by the text

We have seen how we can use LaTeX's marks to get information from the document contents to the headers and footers. The marks mechanism is the only reliable mechanism that you can use to get changing information to the headers or footers. This is because LaTeX may be processing your document ahead before deciding to break the page.

Sometimes the two marks that LaTeX offers are not enough. An example is the following :

> If a solution to an exercise goes across a page break, then I would like to have "(Continued on next page...)" at the bottom of the first page and "(Continued...)" at the top in the margin of the next page.

You cannot use LaTeX's mark mechanisms for this if you also want to use chapter and section information.

The fancyhdr distribution includes a package that gives you two extra marks that can be used in this situation [10]. Here is a way to use this package :

```
\usepackage{extramarks}
...
\pagestyle{fancy}
\lhead{\firstxmark}
\rfoot{\lastxmark}
...
\extramarks{}{Continued on next page\ldots}
Some text that may or may not cross a page boundary...
\extramarks{Continued\ldots}{}
```

Note that the `\extramarks` command must be close to the text, i.e no empty lines (paragraph boundaries) should intervene. Otherwise the page may be broken at that boundary and the extramarks would come on the wrong page.

There are two new marks that can be used in the page layout with this package : If commands of the form `\extramarks{`m_1`}{`m_2`}` are given `\firstleftxmark` [11] gives you the first m_1 value, `\lastleftxmark` gives you the last m_1 value, `\firstrightxmark` gives you the first m_2 value and `\lastrightxmark` gives you the last m_2 value of the current page. Also for convenience (similar to the standard LaTeX marks) `\firstxmark` is an alias for `\firstleftxmark` and `\lastxmark` is an alias for `\lastrightxmark`. There is also a `\topxmark` or `\topleftxmark` similar to TeX's `\topmark` but it probably is not of much use.

Finally it also gives you the `\firstleftmark` and `\lastrightmark` commands that complement the standard LaTeX marks.

To stress the point that marks are the correct way to do this, let me give you a "solution" that will not work [12] :

10. After I made this package I discovered a package `secret.sty` that does a similar thing to mark confidential paragraphs if they cross a page boundary. It does it, however, by changing the output routine.
11. In extramarks.sty before version 2.0 only the shorthand commands `\firstxmark` and `\lastxmark` were defined.
12. Actually there is another way but it requires two LaTeX passes : you can put `\label` commands before and after the text and compare the `\pageref`s.

```
\lhead{Continued}
\rfoot{Continued on next page\ldots}
Some text that may or may not cross a page boundary...
\lhead{}
\rfoot{}
```

You may be tempted to think that the first `\lhead` and `\rfoot` will be in effect when TEX breaks the page in the middle of the text, and the last ones when the page breaks after the text. This is not true as the whole paragraph (including the last definitions) will be processed before TEX considers the page break, so at the time of the page break the last definitions are effective, whether the page break occurs inside the text or outside of it. Putting a paragraph boundary between the text and the last definitions will not work either, because you don't want the first definitions to be in effect when TEX decides to break the page exactly at this boundary. Actually the marks mechanism was invented to get rid of these kinds of problems.

In the above example the text "Continued" appears in the page header. It may be nicer to put it in the margin. This can be easily accomplished by positioning it at a fixed place relative to the page header. In plain TEX you would use a concoction of `\hbox to 0pt`, `\vbox to 0pt`, `\hskip`,`\vskip`, `\hss` and `\vss` but fortunately LATEX's `picture` environment gives a much cleaner way to do this. In order not to disturb the normal header layout we put the text in a zero-sized `picture`. Generally this is the best way to position things on fixed places on the page. You can then also use the normal headings. See also section 21 for another example of this technique.

```
\lhead{\setlength{\unitlength}{\baselineskip}%
\begin{picture}(0,0)
  \put(-2,-3){\makebox(0,0)[r]{\firstxmark}}
  \end{picture}\leftmark}
```

This solution can of course also be used for the footer. Make sure you put the `picture` as the first thing in left-handside entries and last in right-handside ones.

Finally you may want to put "(Continued...)" in the *text* rather than in the header or the margin. Then you have to use the `afterpage` package. We also decide to make a separate environment for it.

```
\newenvironment{continued}{\par
  \extramarks{}{Continued on next page\ldots}
  \afterpage{\noindent\firstxmark\vspace{1ex}}
  }{\extramarks{(Continued\ldots)}{}\par}
```

It is a bit dangerous to use `\firstxmark` outside the page layout routine, but apparently with `\afterpage` this works. If you would need the information further on in the page you must remember the state of the marks in your own variable. You can set this in one of the fancyhdr fields. For example if you want to add something *after* the broken piece of text you can use the following :

```
\newcommand{\mysaved}{}

\newenvironment{continued}{\par
  \extramarks{}{Continued on next page\ldots}
  }{\extramarks{(Continued\ldots)}{}\par\vspace{1ex}\mysaved}
\lhead{\leftmark}
```

```
\chead{\ifthenelse{\equal{\lastxmark}{}}
  {\gdef\mysaved{}}
  {\gdef\mysaved{\noindent[Continued from previous page]}}}
```

If you want to include one of the marks or other varying information in the saved text, you must use `\xdef` rather than `\gdef`.

20 A movie

If you put at each page on the same place a picture that slightly changes from page to page you can get a movie-like effect by flipping through the pages. You can create such a movie easily with fancyhdr. For simplicity we assume that each picture is in a postscript (EPS) file called `pic`⟨n⟩`.ps` where ⟨n⟩ is the page number and that we use the `graphics` or `graphicx` package [13]. To put the movie in the righthandside bottom corner the following will work :

```
\rfoot{\setlength{\unitlength}{1mm}
  \begin{picture}(0,0)
    \put(5,0){\includegraphics{pic\thepage.ps}}
  \end{picture}}
```

Notice that the `\unitlength` parameter should be set locally in the fancyhdr field in order to avoid unwanted interference with its value in the text.

21 Thumb-indexes

Some railroad guides and expensive bibles have so called *thumb-indexes*, i.e. there are marks on the sides of the pages that indicate where the chapters are. You can create these by printing black blobs in the margin of the pages. The vertical position should be determined by the chapter number or some other counter. As the position is independent of the contents of the page, we print these blobs as part of the header in a zero-sized `picture` as described in the previous section.

Of course we have to take care of two-sided printing, and we may want to have an index page with all the blobs in the correct position. The solution requires some hand-tuning to get the blobs nicely spaced out vertically. For the application that I had there were 12 sections, so I made the blobs 18 mm apart, i.e. 9 mm blob separated by 9 mm whitespace. In order to avoid calculations they are set in a `picture` environment with the `\unitlength` set to 18 mm. Page numbers are set in the headers at the outer sides, and the blobs are attached to these. In this example the section numbers are used to position the blobs, but you can replace this with any numeric value. See figure 4 for the resulting overview page and figure 5 for the code.

22 Float placement

Floats are page elements that float with respect to the rest of the document. Standard floats are tables and figures, but with the float package you can easily make new ones, like algorithms. Most of the time floats work satisfactory, but sometimes LaTeX seems too stubborn to do what you want. This section describes how you can influence LaTeX so that it will do most of the time what you

13. If you use an older version of LaTeX you could use the `epsf` or `epsfig` package.

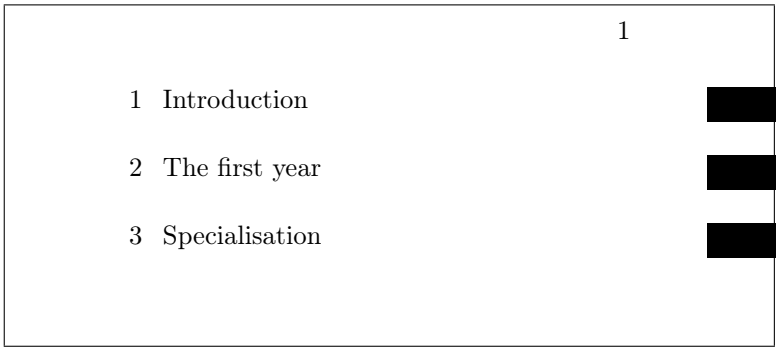


FIGURE 4: Thumb-index overview page

want. There might, however be some pathological cases where it is impossible to convince LaTeX to do things your way. In the following we will use figures as an example but everything applies to other floats as well.

The most encountered problems with floats are :

1. You want a float at a certain position in the text, but LaTeX moves it, usually to the next page.
2. From a certain point, LaTeX moves all your floats to the end of the document or the end of a chapter.
3. LaTeX complains about "Too many floats".

In the first two cases you must first check if you have given the correct "placement" parameter to you float, e.g. `\begin{figure}[htp]` specifies that your figure may be placed either : Here (i.e. in the text position where the command is given), on the Top of a page (which may be the page where you put the command), or on a separate Page of floats. You could also have specified "b" for Bottom of the page. The order of the letters is insignificant, you cannot force LaTeX to try Bottom first and then Top by specifying `[bt]`.

If LaTeX doesn't put the float at the place where you expected it, it is usually caused by the following :

1. The float didn't fit on the page. In this case it has to move to the next page or even further. If you didn't specify either `[t]` or `[b]` in the position parameter, LaTeX must save it until it has enough for a page of floats. So don't specify only `[h]`. If you want to give LaTeX a chance to put the float on a page of floats, you must also specify "p".
2. The placement would violate the constraints imposed by LaTeX's float placement parameters. This is one of the most occurring causes and it can easily be corrected by changing the parameters. Here is a list of them :

Counters – change with `\setcounter`		
`topnumber`	max. number of floats at top of page	2
`bottomnumber`	max. number of floats at bottom of page	1
`totalnumber`	max. number of floats on a page	3
Other – change with `\renewcommand`		
`\topfraction`	max fraction of page for floats at top	0.7
`\bottomfraction`	max fraction of page for floats at bottom	0.3
`\textfraction`	min fraction of page for text	0.2
`\floatpagefraction`	min fraction of floatpage that should have floats	0.5

```
\setlength{\unitlength}{18mm}
\newcommand{\blob}{\rule[-.2\unitlength]{2\unitlength}{.5\unitlength}}

\newcommand\rblob{\thepage
  \begin{picture}(0,0)
    \put(1,-\value{section}){\blob}
  \end{picture}}

\newcommand\lblob{%
  \begin{picture}(0,0)
    \put(-3,-\value{section}){\blob}
  \end{picture}%
  \thepage}

\pagestyle{fancy}
\cfoot{}

\newcounter{line}
\newcommand{\secname}[1]{\addtocounter{line}{1}%
  \put(1,-\value{line}){\blob}
  \put(-7.5,-\value{line}){\Large \arabic{line}}
  \put(-7,-\value{line}){\Large #1}}

\newcommand{\overview}{\thepage
  \begin{picture}(0,0)
    \secname{Introduction}
    \secname{The first year}
    \secname{Specialisation}
...etc...
  \end{picture}}

\begin{document}
\fancyhead[R]{\overview}\mbox{}\newpage % This produces the overview page
\fancyhead[R]{} % Front matter may follow here
\clearpage
\fancyhead[RE]{\rightmark}
\fancyhead[RO]{\rblob}
\fancyhead[LE]{\lblob}
\fancyhead[LO]{{\leftmark}
...
```

FIGURE 5: Thumb-index code

There are also some others for double column floats in two-column documents.

The values in the righthand column are the defaults for the standard LaTeX classes. Other classes could use different defaults. As you see with the default values a float will not be put in the bottom of a page if its height is more than 30% of the page height. So if you specify [hb] for a float which is taller it has to move to a float page. But if it is less than 50% of the page height it will have to wait until some more floats are given before a float page can be filled to satisfy the `\floatpagefraction` parameter. If you have this kind of behaviour you can easily adapt the parameters, e.g. with :

```
\renewcommand{\textfraction}{0.05}
\renewcommand{\topfraction}{0.95}
\renewcommand{\bottomfraction}{0.95}
\renewcommand{\floatpagefraction}{0.35}
\setcounter{totalnumber}{5}
```

You may want to be careful not to make `\floatpagefraction` too small, otherwise you may get too many small floatpages.

You can force LaTeX to ignore most of the parameters for one specific float occurrence by including an exclamation mark (!) in the placement parameters, e.g.

```
\begin{figure}[!htb]
```

Floats which contain a "`t`" in the position parameter could be placed before the place where they are referenced (but on the same page). This is normal behaviour for LaTeX but some people just don't like it. There are a number of ways to prevent this :

1. Of course deleting the "`t`" will help, but in general this is undesirable, as you may want the float to be placed at the top of the next page.
2. use the `flafter` package which causes floats never to be placed "backwards".
3. use the command `\suppressfloats[t]` [14]. This command will cause floats for the top position *on this page* to be moved to the next page. This can also be done with `[b]` or without parameter for all floats on this page.

If in spite of all your attempts LaTeX still moves your floats to the end of the document or the end of a chapter, you can insert a `\clearpage` command. This will start a new page and insert all pending floats before continueing. If it is undesirable to have a pagebreak you can use the afterpage package and the following command :

```
\afterpage{clearpage}
```

This will wait until the current page is finished and then flush all outstanding floats. In some pathological circumstances `afterpage` may give strange results, however.

Finally, if you want a float only at the place where you define it, without LaTeX moving it whatsoever, you can use the float package and give the command :

```
\restylefloat{figure}
```

in the preamble. Now you will be able to specify `[H]` as the position parameter, which will mean "HERE and only HERE". This may cause an unwanted page break however [15]. If you want to

14. This command and the " ! " placement parameter are not defined in LaTeX2.09.
15. There used to be a `here.sty` with the same effect, but this is incompatible with LaTeX 2_ε

avoid the unwanted pagebreak, i.e. let LaTeX move the float only if it doesn't fit on the page, the use the afterpage package with :

```
\afterpage{\clearpage \begin{figure}[H] ... \end{figure}}
```

Complaints from LaTeX about "Too many floats" are usually caused by one of the above problems : floats not being able to be placed and LaTeX collecting too many of them. The solutions given above, especially those with `\clearpage` in them will usually help. In some cases there really are too many floats, as LaTeX has a limited number of "boxes" to store the floats. The package morefloats can be used to increase this number. If you need still more then you must edit a private copy of this file, but even then there will be some limit that you cannot pass. Then your only resort will be to change your document.

23 Multipage Floats

LaTeX's floats cannot be split across pages. Sometimes, however, you want to have a table or figure that doesn't fit on one page. The easiest way is to split these into multiple table or figure environments, but this has a number of undesirable effects :

— Where do you split it ? This is in general more difficult for tables than for figures.
— How do you keep them together ?
— You don't want more than one entry in the list of figures/tables.

Although these problems are not fully solvable in all cases, here are a couple of suggestions :

23.1 Tables

For tables longer than a page you can use the `longtable` package. This package defines a `longtable` environment that is a kind of amalgamation of `table` and `tabular`. It has approximately the same syntax as the `tabular` environment, but it adds some features of `table`, like captions. Longtables will be automatically split when they don't fit on the page. And they will be entered in the list of tables when a caption is given. They will not float, however, and cannot be used inside a float environment. This could mean that another `table` environment, which was defined before the `longtable`, will float past it, and therefore the numbers may get out of order. Another problem could be that the `longtable` starts rather far down the page, which isn't a pleasant sight. If you want the `longtable` to start at the top of the page, the best thing to do is to include it in an `\afterpage` command (using the `afterpage` package). As a `longtable` is by definition large, it is best to put it in a separate file, and `\input` it in the `\afterpage` command :

```
\afterpage{\input{mytable}}

\afterpage{\clearpage\input{mytable}}
```

The last form has the additional advantage that most of the outstanding floats will be printed first.

23.2 Figures

There isn't an equivalent `longfigure` solution, so for figures you will usually have to split it yourself. In general this is less of a problem. However, the problem you get now is how to keep

them together, i.e. how to get the parts on subsequent pages, and how to get a single entry in the list of figures.

You will have to split the figure into pieces and put each part in a separate `figure` environment. To keep them together it is best to use only the `[p]` placement, so that they will be put on floatpages. As they are bigger than a page this is appropriate. The first part would then get a `\caption`, the subsequent parts would be used without a caption. If you want to add a caption-like text, enter it as normal text rather than a `\caption`, so that it will not be entered in the list of figures. It may also be desirable to issue a `\clearpage` first, just like we did for the `longtable`, and to encapsulate this in the `\afterpage` command. E.g.

```
\afterpage{\clearpage\input{myfigure}}
```

where `myfigure.tex` contains :

```
\begin{figure}[p]
\includegraphics{myfig1.eps}
\caption{This is a multipage figure}
\label{fig:xxx}
\end{figure}
\begin{figure}[p]
\includegraphics{myfig2.eps}
\begin{center}
Figure~\ref{fig:xxx} (continued)
\end{center}
\end{figure}
```

You have to make sure that the last part is big enough, otherwise LaTeX could decide to postpone it until it has collected some more floats. This can be done either by making the figure big enough (e.g. by adding some `\vspace`), or by tweaking the `\floatpagefraction` parameter.

If you want your multipage figure to start at a lefthand-side (even-numbered) page you can use a test in the `\afterpage` command (using the `ifthen` package) :

```
\afterpage{\clearpage
\ifthenelse{\isodd{\value{page}}{\afterpage{\input{myfigure}}} % odd page
    {\input{myfigure}}}} % even page
```

If there are too many floats on the skipped page, this may still fail to start your multipage figure on an even page, however.

24 Contact information

Piet van Oostrum
Dept. of Computer Science
Utrecht University
P.O. Box 80.089
3508 TB Utrecht, The Netherlands
Telephone : +31 30 2532180 Telefax : +31 30 2513791
E-mail : piet@cs.uu.nl
WWW : http ://www.cs.uu.nl/people/piet

Index