The \texttt{varioref} package\footnote{This file has version number v1.2c, last revised 1999/12/02.}

Frank Mittelbach

1999/12/02

Abstract

This package defines the commands \texttt{\vref, \vpageref, \vrefrange,} and \texttt{\vpagerefrange} for \LaTeX\,2ε. \texttt{\vref} is similar to \texttt{\ref} but adds an additional page reference, like ‘on the facing page’ or ‘on page 27’ whenever the corresponding \texttt{\label} is not on the same page. The command \texttt{\vpageref} is a variation to \texttt{\pageref} with a similar functionality. The \texttt{\v...range} commands take two labels as arguments and produce strings which depend on whether or not these labels fall onto a single page or on different pages. Generated strings are customizable so that these commands are usable with various languages.

1 Introduction

In many cases it is helpful when refering to a figure or table to put both a \texttt{\ref} and a \texttt{\pageref} command into the document especially when there are one or more pages between the reference and the object. Therefore some people use a command like

\begin{verbatim}
\newcommand{\fullref}[1]{\ref{#1} on page~\pageref{#1}}
\end{verbatim}

which reduces the number of key strokes, necessary to make such a complete reference. But since one never knows where the referenced object finally falls, using such a device may result in a page reference to the current page which is disturbing and therefore should be avoided.

2 The user interface

\texttt{\vref} The implementation of \texttt{\vref} below produces only a \texttt{\ref} when reference and \texttt{\label} are on the same page. It will additionally produce one of the strings ‘on the facing page’, ‘on the preceding page’, or ‘on the following page’, if label and reference differ by one and it will produce both \texttt{\ref} and \texttt{\pageref} when the difference is larger. The word ‘facing’ is used when label and reference both fall onto a double spread. However, if a special page numbering scheme is used instead
of the usual arabic numbering (e.g., \pagenumbering{roman}) then there will be no distinction between one or many pages off.

Sometimes one wants to refer only to page number and again such a reference should normally be suppressed if we are referring to the current page. For this purpose the package defines the \vpageref command. It will produce the same strings as \vref except that it doesn't start with the \ref and except that it will produce the string that is saved in \reftextcurrent if label and reference fall onto the same page. By defining \reftextcurrent to produce “on this page” or something similar, we can avoid that

... see the example \vpageref{ex:foo} which shows ...

comes out as “… see the example which shows …”, which could be misleading.

You can put a space in front of \vpageref it will be ignored if the command doesn’t produce any text at all.

But in fact \vpageref allows even more control. If has two optional arguments. With the first one, one can specify the text that should be used if label and reference fall on the same page. This is very helpful if both are near to each other, so that they may or may not be separated by a page break. In such a case we usually know (!) whether the reference is before or after the label so that we can say something like

... see the example \vpageref[above]{ex:foo} which shows ...

which will then come out as “… see the example above which shows …” if we are still on the same page, but as “… see the example on the page before which shows …” (or something similar depending on the settings of the \reftext commands) when there was a page break in the meantime.

One warning however, if you use \vpageref with the optional argument to refer to a figure or table, keep in mind that depending on the float placement parameters the float may show up on top of the current page and therefore before the reference even if it came after it in the source file.

But maybe you prefer to say “… see the above example” if example and reference fall onto the same page, i.e., reverse the word order. In fact, in some languages the word order automatically changes in that case. To allow for this variation the second optional argument can be used. It specifies the text preceding the generated reference if object and reference do not fall onto the same page. Thus one would write

... see the \vpageref[above example]{ex:foo} which shows ...

to achieve the desired effect.

2.1 Additions in 1998

\vrefrange This command is similar to \vref but it takes two mandatory arguments denoting
a range to refer to (e.g., a sequence of figures or a sequence of equations, etc.). So if \texttt{fig:a} is your first figure in the sequence and \texttt{fig:c} your last you can write

\begin{verbatim}
... see figures \vrefrange{fig:a}{fig:c} ...
\end{verbatim}

which would then be formatted as

\begin{verbatim}
... see figures 3.4 to 3.6 on pages 23–24 ...
\end{verbatim}

or, if they happen to all fall onto the next page, as

\begin{verbatim}
... see figures 3.4 to 3.6 on the following page ...
\end{verbatim}

i.e., the command is deciding what to say depending on where the two labels are placed in relation to each other; it is essentially implemented using \texttt{\vpagerefrange} described below. The optional argument the command may take is the text to use in case both labels are placed on the current page.

\texttt{\vpagerefrange} This command is similar to \texttt{\vpageref} but takes two mandatory arguments which are two labels denoting a range. If both labels fall onto the same page, the command acts exactly like \texttt{\vpageref} (with a single label), otherwise it produces something like “on pages 15–18” (see customization possibilities below). The optional argument it may take is the text to use in case both labels are placed on the current page.

\texttt{\vrefpagenum} This macro is provided to allow the user to write their own small commands which implement functions similar to those provided by the two previous commands. It takes two arguments: the second is a label (i.e., as used in \texttt{\label} or \texttt{\ref}) and the first is an arbitrary command name (make sure you use our own) that receives the page number related to this label. So if you have two (or more) labels you could retrieve their page numbers, compare them and then decide what to print. For example, the following not very serious definition (also using the \texttt{ifthen} package)

\begin{verbatim}
\newcommand{\amusingversion}[2]{% 
  \vrefpagenum{\firstnum}{#1}% 
  \vrefpagenum{\secondnum}{#2}% 
  the definition% 
  \ifthenelse{\equal{\firstnum}{\secondnum}}% 
    {s of \ref{#1} and \ref{#2} \vpageref{#1}}% 
    { of \ref{#1} \vpageref{#1} and of \ref{#2} \vpageref{#2}}% 
  }
\end{verbatim}

\begin{verbatim}
...\amusingversion{foo}{bar}
\end{verbatim}

will print something like

\begin{verbatim}
... the definitions of 3 and 4 on the previous page
\end{verbatim}

in the case both labels are on the same page but something like

\begin{verbatim}
... the definition of 3 on the next page and of 4 on page 13
\end{verbatim}

in case the are on different pages.
3 Customization

The package supports all options defined by the babel package to translate the fixed strings into other languages than English. (Some languages need updating, however.) It also supports languages currently not in babel; check the section on options later on. You can also modify some or all of the strings by redefining the following commands. Backward references use \reftextbefore if the label is on the preceding page but invisible and \reftextfacebefore if it is one the facing page (i.e., if the current page number is odd). Similarly \reftextafter is used when the label comes on the next page but one has to turn the page and \reftextfaceafter if it is on the following but facing page.

In fact, \reftextface... is used only if the user or the document class specified two-sided printing.

Finally we have \reftextfaraway which is used whenever label and reference differ by more than one or when they aren’t numeric. This macro is a bit different because it takes one argument, the symbolic reference string so that one can make use of \pageref in its replacement text.

To allow a bit random variation in the generated strings one can use the command \reftextvario inside the string macros. It takes two arguments and selects one or the other for printing depending on the number of already seen \vref or \vpageref commands. As an example see the definitions of \reftextbefore etc. on page 7.

3.1 Additions in 1998

The commands \vrefrange and \vpagerefrange produce their text using two macros described below. By redefining them one can modify the results to accommodate special requirements.

They both take two mandatory arguments denoting the first and the last label of the range.

This macro produces text that describes the the page range of the two labels, e.g., the default for English is “on pages\pageref{#1}--\pageref{#2}”.

This macro produces text that describes the range of figures, tables, or whatever the labels refer to, the default for English is “\ref{#1} to\ref{#2}”.

4 Options

As mentioned above the package supports all standard options offered by the Babel system to customize the strings produced. In addition it offers the option draft to turn error messages into warnings during development. The default final produces error message when a generated string falls onto a page boundary (see next section).
5 A few warnings

Defining commands like the ones described above poses some interesting problems. Suppose, for example, that a generated text like ‘on the next page’ gets broken across pages. If this happens it is very difficult to find an acceptable solution and in fact can even result in a document that will always change from one state to another (i.e., inserting one string, finding that this is wrong, inserting another string on the next run which makes the first string correct again, inserting …). The current implementation of \texttt{varioref} therefore issues an error message whenever the generated text is broken across page boundaries, e.g.,

\begin{verbatim}
\vrefwarning
\vrefshowerrors
\end{verbatim}

\begin{verbatim}
table 5 on the current ⟨page break⟩ page
\end{verbatim}

would result in an error, which needs to be resolved by the user by replacing the \texttt{\vref} command with an ordinary \texttt{\ref} just before the final run. This is not completely satisfactory but in such case no solution really is. During document preparation, while one is still changing the text, such error messages can be turned into warnings by placing a \texttt{\vrefwarning} command in the preamble. This is equivalent to specifying “draft” as an option to the package. \texttt{\vrefshowerrors} ensures that varioref stops when detecting a possible loop. This is the default and equivalent to specifying “final” as an option.

At the end final a warning: every use of \texttt{\vref} will internally generate two macro names to keep track of the string positions within the document. As a result you may run out of name space or main memory if you make heavy use of this macro on a small \TeX{} installation. For this reason the primitive command \texttt{\fullref} is also provided. This command can be used whenever you know for sure that label and reference can’t fall onto nearby pages.

6 The documentation driver file

The next bit of code contains the documentation driver file for \TeX{}, i.e., the file that will produce the documentation you are currently reading. It will be extracted from this file by the \texttt{docstrip} program.

\begin{verbatim}
1 ⟨∗driver⟩
2 \documentclass{ltxdoc}
3 \usepackage{varioref}
4 \GetFileInfo{varioref.sty}
5 \setlength\hfuzz{1pt} % ignore slight overfulls
6 \EnableCrossrefs
7 \DisableCrossrefs % Say \DisableCrossrefs if index is ready
8 \RecordChanges % Gather update information
9 \OnlyDescription % comment out for implementation details
10 \begin{document}
11 \DocInput{varioref.dtx}
12 \end{document}
13 ⟨/driver⟩
\end{verbatim}
The implementation

The main implementation idea is to generate an internal label command for every `\pageref` and compare the page reference of this label with the page reference of the user-requested label. Since this means one additional macro name for each use of `\pageref` or `vref` (which is implemented in terms of `\pageref`) the memory requirements of this package may be high in terms of main and macro name memory. Since the information held in the internal labels is used only once one could envision a different scheme if suitable primitives for this kind of “one place information” would be provided by \LaTeX{}X. This type of data structure is planned for \LaTeX{}X3 but for the moment we have to live with the memory restrictions.

We start by checking if this file was already loaded. If not we identify the current version. This is actually done at the top of the file, so we comment it out here.

To support the use of babel we want to add the additional strings to the `\extras{language}` commands. Since `\addto` in the current implementation of babel has a bug that does not allow to use arguments containing hashmarks we do this by hand.

Excuse that we don’t know the strings for a certain language.
If you can suggest translations for this language, please mail them to the author of this package.

\vref@stringwarning And we sometimes just don't know some strings ...

\def\vref@stringwarning#1{\PackageWarningNoLine{varioref}{Sorry, there is no proper translation for the string produced by \protect#1. English string used instead.\PackageWarningNoLine{varioref}{If you can suggest translations for the current language, please mail them to the author of this package}}}

\reftextfaceafter \reftextfacebefore \reftextbefore \reftextcurrent The options do set the macros that generate the textual strings. Note, that they do not start with a space, the space is already added in the main macro below.

\DeclareOption{american}{% 
\vref@addto\extrasamerican{% 
\def\reftextfaceafter {on the \textit{facing} next page} 
\def\reftextfacebefore {on the \textit{facing} preceding page} 
\def\reftextafter {on the \textit{following} next page} 
\def\reftextbefore {on the \textit{preceding} page before} 
\def\reftextcurrent {on \textit{this} page} 
\def\reftextfaraway#1{on page~\pageref{#1}} 
\def\reftextpagerange#1#2{on pages~\pageref{#1}--\pageref{#2}} 
\def\reftextlabelrange#1#2{\ref{#1} to \ref{#2}} 
}}

The austrian defaults are the same as the german ones.

\DeclareOption{austrian}{% 
\vref@addto\extrasaustrian{% 
\def\reftextfaceafter {auf der n"achsten Seite} 
\def\reftext.facebefore {auf der vorherigen Seite} 
\let\reftextafter \reftextfaceafter 
\let\reftextbefore \reftextfacebefore 
\def\reftextcurrent {auf dieser Seite} 
\def\reftextfaraway#1{auf page~\pageref{#1}} 
\def\reftextpagerange#1#2{auf Seiten~\pageref{#1}--\pageref{#2}} 
\def\reftextlabelrange#1#2{\ref{#1} bis \ref{#2}} 
}}

Text for brazil defaults was contributed by Alcino Dall Igna Junior (adij@di.ufpe.br).
Text for Breton defaults was contributed by Christian ROLLAND
(Christian.Rolland@univ-brest.fr).

Text for Catalan defaults was contributed by Robert Fuster (rfuster@mat.upv.es).
Defaults for Danish provided by Torsten Martinsen (tmart91@kom.auc.dk).

Default string for Dutch have been contributed by Frank Poppe (POPPE@SWOV.NL). This option currently supports one additional string macro \refpagename so that you can easily change to bladzijde instead of pagina if you prefer this word for
“page”. However, I will not guarantee that this will survive future versions of this package, so use it on your own risk (you can always update the full strings to be on the safe side).

\DeclaReOption{dutch}
\{\vref@addto\extrasdutch{\%
  \def\refpagename{pagina}\
  \def\reftextfaceafter {op de \reftextvario{rechter ~\refpagename}\
  {\refpagename} hiernaast}}\
  \def\reftextfacebefore{op de \reftextvario{linker ~\refpagename}\
  {\refpagename} hierna}}\
  \def\reftextafter {op de \reftextvario{volgende ~\refpagename}\
  {\refpagename} hierna}}\
  \def\reftextbefore {op de \reftextvario{vorige ~\refpagename}\
  {\refpagename} hiervoor}}\
  \def\reftextcurrent {op deze ~\refpagename}}\
  \def\reftextfaraway#1{op \refpagename~\pageref{#1}}\
  \def\reftextpagerange#1#2{op pagina's~\pageref{#1} t/m~\pageref{#2}}\
  \def\reftextlabelrange#1#2{\ref{#1} t/m~\ref{#2}}}\

\DeclaReOption{english}
\{\vref@addto\extrasenglish{\%
  \def\reftextfaceafter {on the \reftextvario{facing}{next} page}\
  \def\reftextfacebefore{on the \reftextvario{facing}{preceding}\
  page}}\
  \def\reftextafter {on the \reftextvario{following}{next} page}\
  \def\reftextbefore {on the \reftextvario{preceding page}{page\
  before}}}\
  \def\reftextcurrent {on \reftextvario{this}{the current} page}\
  \def\reftextfaraway#1{on page~\pageref{#1}}\
  \def\reftextpagerange#1#2{on pages~\pageref{#1}--\pageref{#2}}\
  \def\reftextlabelrange#1#2{\ref{#1} to~\ref{#2}}}\

\DeclaReOption{esperanto}
\{\vref@addto\extrasesperanto{\%
  \def\reftextfaceafter {on the \reftextvario{facing}{next} page}\
  \def\reftextfacebefore{on the \reftextvario{facing}{preceding}\
  page}}\
  \def\reftextafter {on the \reftextvario{following}{next} page}\
  \def\reftextbefore {on the \reftextvario{preceding page}{page\
  before}}}\
  \def\reftextcurrent {on \reftextvario{this}{the current} page}\
  \def\reftextfaraway#1{on page~\pageref{#1}}\
  \def\reftextpagerange#1#2{on pages~\pageref{#1}--\pageref{#2}}\
  \def\reftextlabelrange#1#2{\ref{#1} to~\ref{#2}}}\

\DeclaReOption{finnish}
\{\vref@addto\extrafinnish{\%
  \def\refpagename{pagina}\
  \def\reftextfaceafter {op de \reftextvario{rechter ~\refpagename}\
  {\refpagename} hiernaast}}\
  \def\reftextfacebefore{op de \reftextvario{linker ~\refpagename}\
  {\refpagename} hierna}}\
  \def\reftextafter {op de \reftextvario{volgende ~\refpagename}\
  {\refpagename} hierna}}\
  \def\reftextbefore {op de \reftextvario{vorige ~\refpagename}\
  {\refpagename} hiervoor}}\
  \def\reftextcurrent {op deze ~\refpagename}}\
  \def\reftextfaraway#1{op \refpagename~\pageref{#1}}\
  \def\reftextpagerange#1#2{op pagina's~\pageref{#1} t/m~\pageref{#2}}\
  \def\reftextlabelrange#1#2{\ref{#1} t/m~\ref{#2}}}\

\DeclaReOption{finnish}
\{\vref@addto\extrafinnish{\%
  \def\refpagename{pagina}\
  \def\reftextfaceafter {op de \reftextvario{rechter ~\refpagename}\
  {\refpagename} hiernaast}}\
  \def\reftextfacebefore{op de \reftextvario{linker ~\refpagename}\
  {\refpagename} hierna}}\
  \def\reftextafter {op de \reftextvario{volgende ~\refpagename}\
  {\refpagename} hierna}}\
  \def\reftextbefore {op de \reftextvario{vorige ~\refpagename}\
  {\refpagename} hiervoor}}\
  \def\reftextcurrent {op deze ~\refpagename}}\
  \def\reftextfaraway#1{op \refpagename~\pageref{#1}}\
  \def\reftextpagerange#1#2{op pagina's~\pageref{#1} t/m~\pageref{#2}}\
  \def\reftextlabelrange#1#2{\ref{#1} t/m~\ref{#2}}}\

The finnish strings were suggested by Matti Rintala (bitti@cs.tut.fi).
French defaults are provided by Daniel Flippo (Daniel.Flipo@univ-lille1.fr).

Galician defaults are provided by Matthias Moebius (Matthias.Moebius@uni-konstanz.de).

There are no good variants for German (I think and still think but this is a matter of taste :-). The following definitions were recently suggested to me but since the original are in for a long time i don’t want to change them now since that
could make a lot of documents change their formatting. If you fancy them, add a
redefinition of the corresponding macro(s) to the preamble of your document.

\def\reftextfaceafter {auf der \reftextvario
{gegen"uberliegenden}{anderen} Seite}\
\def\reftextfacebefore {auf der \reftextvario
{gegen"uberliegenden}{anderen} Seite}\
\def\reftextafter {auf der \reftextvario
{n"achsten}{folgenden} Seite}\
\def\reftextbefore {auf der \reftextvario
{vorigen}{vorhergehenden} Seite}\
\def\reftextcurrent {\reftextvario
{auf dieser}{diese} Seite}\

\DeclareOption{german}
  {\vref@addto\extrasgerman{\
    \def\reftextfaceafter {auf der n"achsten Seite}\
    \def\reftextfacebefore {auf der vorherigen Seite}\
    \let\reftextafter \reftextfaceafter\
    \let\reftextbefore \reftextfacebefore\
    \def\reftextcurrent {auf dieser Seite}\
    \def\reftextfaraway#1{auf Seite~\pageref{#1}}\
    \def\reftextpagerange#1#2{auf Seiten~\pageref{#1}--\pageref{#2}}\
    \def\reftextlabelrange#1#2{\ref{#1} bis~\ref{#2}}}}

Defaults for greek suggested by Apostolos Syropoulos
(apostolo@obelix.ee.duth.gr).

\DeclareOption{greek}
  {\vref@addto\extrase\%\
    \def\reftextfaceafter {sthn \reftextvario{paro'usa}\
    {ep'omenh} sel'ida}\
    \def\reftextfacebefore{sthn \reftextvario{paro'usa}\
    {prohgo'umenh} sel'ida}\
    \def\reftextafter {sthn ep'omenh sel'ida}\
    \def\reftextbefore {sthn prohgo'umenh sel'ida}\
    \def\reftextcurrent {s'' aut'h th sel'ida}\
    \def\reftextfaraway#1{sth sel'ida\nobreakspace\pageref{#1}}\
    \def\reftextpagerange#1#2{stis sel'ides\nobreakspace\pageref{#1}---\pageref{#2}}\
    \def\reftextlabelrange#1#2{\ref{#1} ws\nobreakspace\ref{#2}}}}

Defaults for Italian suggested by Giovanni Pensa (pensa@dsi.unimi.it) with
\TeXnici.

\DeclareOption{italian}
  {\vref@addto\extrais\%\
    \def\reftextfaceafter {\reftextvario{a fronte}\
    {nella pagina successiva}}\
    \def\reftextfacebefore{\reftextvario{a fronte}\
    {nella pagina precedente}}}
Defaults for Hungarian by Jeff Goldberg (J.goldberg@Cranfield.ac.uk). There is a problem with the use of the definite article a or az in Hungarian before expansions of \ref or \vref. The problem is that the the word should be az if the number following would begin with a vowel if spelled out, and a if the number would begin with a consonant.

The option assumes that there is a command \aza defined which helps resolving this problem.\footnote{This information is actually quite old but unfortunately never made it into varioref. In case any work has been undertaken to resolve this please inform the author about it.}

The Norwegian and ‘nynorsk translations have been provided by Dag F Langmyhr <dag@ifi.uio.no>.

The information is actually quite old but unfortunately never made it into varioref. In case any work has been undertaken to resolve this please inform the author about it.
The polish translations have been provided by Tomasz Michniewski <tomekm@ias.wat.waw.pl>. The use of the command \eob and aob etc. requires appropriate definitions as provided, for example, by the babel system. This should probably be handled differently but it would require modifications in babel’s language support.

The portuguese default text has been provided by Jose Carlos Oliveira Santos jcsantos@fc.up.pt <tomekm@ias.wat.waw.pl>.
The text for Russian was taken from the book in Russian by I. A. Kotelnikov and P. Z. Chebotaev, and adapted to the encoding-independent form by Vladimir Volovich (vvv@vvv.vsu.ru).
Text for Spanish defaults was contributed by Julio Sanchez (jsanchez@gmv.es).

The text for Swedish was contributed by Mats Dahlgren (matsd@sssk.se).
During document preparation errors due to \vref are usually of not much interest. For this reason the draft option turns them into warnings by issuing a \vrefwarning command. The final option (default) stops with an error message when varioref detects a possible loop.

\DeclareOption{draft}{\AtEndOfPackage\vrefwarning}
\DeclareOption{final}{\AtEndOfPackage\vrefshowerrors} % the default
\ExecuteOptions{draft}\ExecuteOptions{final}

7.2 Defining the main macros

\vr@cnt

To generate new labels we use a counter and construct the internal label names by prefixing the current counter value with the string vr@. We need a LaTeX counter to get \includes handled correctly.
\newcounter{vrcnt}
The command \vpageref generates the appropriate string by first parsing the optional arguments (if any) and then compares the internal and the user label. This command should be robust since the user might put it anywhere.

\DeclareRobustCommand\vpageref{\@ifnextchar[\
\@vpageref{\@vpageref\reftextcurrent}}%

The default for the first optional argument is the string hidden in the macro \reftextcurrent.

\def\@vpageref[#1]{\@ifnextchar[\
{\@@vpageref{#1}}{\@@vpageref{#1}\unskip\space}}%

The default for the second optional argument is a space which is prefixed by \unskip to get rid of any leading space inserted already.

\@@vpageref The \@@vpageref macro finally generates the references by comparing the value of an internal label with the value of the user label.

\def\@@vpageref#1[#2]#3{\leavevmode\unskip
First it switches to horizontal mode if necessary and also removes any leading space.

\global\advance\c@vrcnt\@ne
Then it checks if for the current value of \c@vrcnt a label command was issued in the last run. If not it pretends that there was one with the value {??}{??}. Thereafter it stores the the pageref value for this label in the macro \@tempa.

\vref@pagenum\@tempa{\the\c@vrcnt @vr}
Here we check that the generated text is not going across a page boundary.

\vref@pagenum\@tempb{\the\c@vrcnt @xvr}
In version 1.0l the label string has been turned around so that the number comes first. This was done to allow easy explicit expansion of the number before it is passed to the label command. In the babel system the argument of label was not expanded with the result that wrong label strings have been generated. This is a general problem that might need a completely different solution in babel but for now the change below (and in some other places in the code) should solve the problem for this special combination of packages.

\expandafter\label\expandafter{\the\c@vrcnt @xvr}
Inside displays of the amsmath package the label command is redefined which makes the test for loops incorrect if a vref is used in, say, \intertext. So we test this condition first and only do the test if label doesn’t have a special meaning.
If both points do not fall onto the same page with either issue an error or a warning message.

\vref@err{\noexpand\vref at page boundary
\@tempb-\@tempa\space (may loop)\%}
\fi
\fi

Same game for the user requested label; this time the page reference is saved in \@tempb.

\vrefpagenum\@tempb{#3}\%

Now after the internal label has served its purpose if would be nice to free the memory it occupies by using something like

\global\expandafter\let\csname r@\the\c@vrcnt @vr\endcsname\@empty

But this is not possible because it would result in getting ‘Labels may have changed...’ warnings for every run. Now we are ready to produce the textual strings. Since we have removed any leading space we now insert a space and then compare the two page references.

\space
\ifx\@tempa\@tempb
\#1%
\else
\#2%
\fi

Now we check if the page number of the referenced object (stored in \@tempb is a single positive number.

\is@pos@number\@tempb
\%

If so, we check if the current position (stored in \@tempa) is a positive number. If this is the case we assign this number to the counter register \@tempcnta and add one to it.

\is@pos@number\@tempa
\{\@tempcnta\@tempa
\advance\@tempcnta\@ne
\%

If it is not a positive number we assign the largest possible number to \@tempcnta and thereby pretending that label and reference are miles away from each other.

\{\@tempcnta\maxdimen\%

Now we are ready to check if reference and object are on nearby pages. \@tempb will expand to the page number of the object (and we know that this is a number) and \@tempcnta is either one higher than the reference page or completely out of
bounds. So if both represent the same value then the object lies one page after
its reference.

516 \ifnum \@tempb =\@tempcnta

Thus if the object falls onto an odd page then the reference is on the facing even
page (and so we insert \reftextfaceafter), otherwise the object can not be seen
from the reference (and we in insert \reftextafter). Don’t be surprised if we
are not using \@tempb in the check. Since \@tempcnta has the same value it is
faster to use the register instead of parsing the macro contents anew.

517 \ifodd\@tempcnta

In fact we are going to use \reftextfaceafter only if we are doing two-sided
printing, otherwise \reftextafter is always used. Since the value of \if@twoside
is evaluated before reading in packages we could do better (saving some tokens)
by defining the current macro in dependence of this boolean.

518 \if@twoside
519 \reftextfaceafter
520 \else
521 \reftextafter
522 \fi
523 \else
524 \reftextafter
525 \fi

If the object is not on the page following the reference we check if it is on the
page before the reference. In \LaTeX{} this situation is not too common, for ex-
ample with floats it normally does not occur, but of course it isn’t impossible if
you more than one reference to the same object, or if you have back references
to sections, theorems, etc. To test this we now substract two from the current
value of \@tempcnta (which was set to one higher as the reference page number).
Note, that substraction is also possible if the value was \maxdimen — we still get
something that is much larger than any sensible page number.

526 \else
527 \advance\@tempcnta-2

If now \@tempb and \@tempcnta have the same value then the object lies one page
before the reference.

528 \ifnum \@tempb =\@tempcnta

Again we have to check for odd or even pages to distinguish between the facing
and the non-facing situation.

529 \ifodd\@tempcnta
530 \reftextbefore
531 \else
532 \if@twoside
533 \reftextfacebefore
534 \else
535 \reftextbefore

20
If the above test also returns false then we have object and references on pages which are far away or don’t contain simple page numbers. Therefore we generate the \reftextfaraway string. Recall that this is a macro which has the user label as an argument.

\fi
\fi

We do the same if our first test (that the page with the object has a positive page number) turns out to be false.

\else
\reftextfaraway{#3}%%
\fi
\fi

Finally we generate the internal label so that it can be check on the next run. This means that we compare the position after the string with the position of the referenced object. There is one thing to note: to conserve space we locally make \@currentlabel empty since we are only interested in the page number value of this internal label.

{\let\@currentlabel\@empty
The setting of \df@label is a concession to the amsmath package which might redefine \label and expects this macro to be empty in certain circumstances.

\let\df@label\@empty
\expandafter\label\expandafter{\the\c@vrcnt @vr}}%%

When the \vref command detects a possible crossing over a page boundary it will call \vref@err to generate an error message. During document preparation one can turn such errors into warnings by issuing a \vrefwarning declaration.

\def\vrefwarning{\def\vref@err{\PackageWarning{varioref}}}
\def\vrefshowerrors{\def\vref@err##1{\PackageError{varioref}{##1}{Please check the pages in question. You might need to replace the \string\vref\MessageBreak or \noexpand\vpageref by a normal \noexpand\(page)ref to stop LaTeX running forever.}}}

The use of \nobreakspace or ~ after the \unskip means that this command will always produce a normal space while something like \nobreak\space will react to settings of \nonfrenchspacing. Unfortunately the latter will also act in cases where one really wants a normal space, e.g., in Fig.~\vref{..}, resulting in a large extra space after the dot. For this reason the first solution is used.
The utility macro \is@pos@number takes three arguments: a string that is tested for being a valid integer and the actions to be taken in case the test comes out true or false.

We start by passing the string to the macro \is@pos@num after prefixing it with a 0 and adding a space after it. To have a well defined ending point we also add \@nil at the end.

\def\is@pos@number#1{\is@pos@num0#1\space\@nil}

Now we parse the enlarged string into a counter register. To get control back after everything that is regarded by \TeX as a number is put into that register we assign \is@pos@num@ to the \afterassignment token. Since the string started with 0, we can be sure that the register assignment will be carried out without an error message. In case of counter assignments leading zeros are discarded.

\def\is@pos@num{\afterassignment\is@pos@num@ \count@}

Now we have to check whether or not the whole string was parsed into that register or some remainder was left over. Since we have added a \@nil token at the very end we can use that to delimit the argument of \is@pos@num@. Note, that the added space in \is@pos@number above gets parsed away by the counter assignment. If it would be missing, and the full string would consist of a number, \TeX would try to replace \@nil by its definition to see if it would contain additional digits and thus we would be in trouble at this point.

\def\is@pos@num@#1\@nil{\if0#10\%

Depending on the result we execute the second or third argument of the command \is@pos@number using a technique that removes the \else or \fi from the input stream first, so that in principle input stream parsing could be done from within the arguments.

\expandafter\@firstoftwo \else
\expandafter\@secondoftwo \fi}

This macro gives a little bit of random variation in the text because the outcome depends on the number of \vref commands seen before.

\def\refextvario#1#2{\ifodd\c@vrcnt #1\else#2\fi}

And here is the primitive command that always produces a \ref and a \pageref.

\def\fullref#1{\ref{#1} \refextfaraway{#1}}
\subsection{Supporting ranges}

\vref@pagenum This command takes a label as #1 extracts the page number associated with it and saves the result in a csname which is passed as #1. If the label isn’t associates with a page number it essentially associates it with ?? for both the label number and the page, thus defining the label this way. This means that no label warning is generated for this label in case it is undefined.

\vrefpagenum Same as the above but this time we generate a warning for undefined labels.

\vpagerefrange If both labels refer to the same page then we produce a \vpageref. Otherwise we produce the result of \reftextpagerange. One could think of making this command even more clever by producing a special string if the difference between the two page number is one (but this is not done).

\vrefrange

\vrefrange