Grzegorz Murzynowski

The gmverse Package[*]

September 4, 2008

This is (a documentation of) file gmverse.sty, intended to be used with LaTeX 2_ε. This package redefines verse environment to optionize \\ for line ends and gives it a possibility of optical centering and 'right-hanging' alignment of lines broken because of length.

Written by Natror (Grzegorz Murzynowski),
natror at o2 dot pl
© 2006, 2008 by Natror (Grzegorz Murzynowski).
This program is subject to the LaTeX Project Public License.
See <http://www.ctan.org/tex-archive/help/Catalogue/licenses.lppl.html>
        for the details of that license.
LPPL status: "author-maintained".
Many thanks to my TeX Guru Marcin Woliński for his TeXnical support.

```
62 \NeedsTeXFormat{LaTeX2e}
63 \ProvidesPackage{gmverse}
64      [2008/09/04␣v0.73␣A␣redefinition␣of␣verse␣env.␣(GM)]
```

Contents

Intro, General Usage

An inspiration for me was the verse package by Peter Wilson and Herries Press but, as you will see, I take a slightly different task. The main goal is to make verse typeset long and broken lines flush right (i.e., the beginning line flush left as usual and the ending one flush right) and optionally gives the ending a square bracket ([), as sometimes practiced in Polish typesetting of poetry. This main task is fulfilled by \versehangright declaration executed by default and the [option is provided by \versehangrightsquare declaration, executed if the package is loaded with squarebr option.

\versehangright
\versehangrightsquare
squarebr
\verseopticalcenter

\vocweigh

One more declaration is \verseopticalcenter that makes verses centered due to their baric centre. I gently deprecate this declaration since it requires also another, \defvocgamma[true] or [false], and what you need most probably is \vocweigh3—

---

[*] This file has version number v0.73 dated 2008/09/04.

a declaration of optical centering due to the sum of cubics of the line lengths (it looks best) and declares all what's needed. If you type \vocweigh[]3, the long lines will be broken with [preceding the lower part.

My (little) experience is with typesetting great poetry the power of which is not in the shape of verses and stanzas but in the words so I did not pay much attention such things as sophisticated indentation. And my opinion is that all verses in one document (book) should be typeset as similarly as possible so that the power of their words could rather talk not the variety of their shapes. That's why my redefinition of verse is done at the beginning and once forever.

local But if someone *really* has to use different versions of verse, I left him a doorway: the local option of the package, with which the declarations redefining verse are not executed (and, of course, may be put anywhere the user wishes, with all the scoping rules). If local option is used, there's no use of squarebr option and the latter does no harm.

* Another task I set myself was to "preserve" known commands so that they work in a known way, namely *[⟨*skip*⟩] and blank line. But in these redefined verse's one isn't forced to end every line with \\. Ending a line with just the ^^M character will work too.

Parameters

As you can see below, my verse, unlike the LaTeX's and verse's ones, does not create a list environment. It uses the following parameters:

> \verseskipbefore \verseskipafter \stanzaskip \betweenstanzaspenalty
> \instanzapenalty \versegenericlinewidth \versemaxlinewidth

I suppose their names are self-explaining. Maybe except the last two. \versegeneri-
\versegenericlinewidth clinewidth is a dimen register intended to store an average line width of a verse. It'll be used to set the optical centering.

What optical centering is? The basic assumption of this package is that all the verses are aligned left (ragged right). Optical centering of a verse is such setting the left margin that the verse (a ragged body of text) seems to be balanced at the vertical axis of the page. In other words, to human eyes the baric centre of the verse seems to lie on the vertical axis of the page.

\versemaxlinewidth The \versemaxlinewidth is a dimen register to store a maximum acceptable line width. The lines exceeding this width will be broken and (if the default declaration \versehangright(square) is in force) the broken ending will be aligned right.

\versecenterdue
\versemaxline
\ifcenterverse
\ifmaxversewidth
Those two parameters should only be set with \versecenterdue{⟨*generic line (text)*⟩} and \versemaxline{⟨*longest acceptable line (text)*⟩} commands, which set the dimens *and* switch proper switches, \ifcenterverse and \ifmaxversewidth respectively.

If you have used \versecenterdue or \versemaxline and since some point you don't want center the verse due to generic line width (n)or delimit line's width, just write \centerversefalse or \maxversewidthfalse respectively.

\versecenterdue takes one argument, which should be any text, rule or \hskip of desired width. The starred version \versecenterdue* takes explicit width (dimen) as its argument.

\versemaxline works the same way.

\betweenstanzaspenalty \betweenstanzaspenalty is the penalty put between the stanzas, as its name suggests. Its goal is to encourage the pagebreaks between two stanzas not in the middle

of a stanza. By default, it is set equal `\interlinepenalty` at `\begin{document}` but thanks to a little trick you can set it also in the preamble[1].

`\instanzapenalty`   `\instanzapenalty` is the penalty but between two lines of a stanza. Its goal is to discourage the pagebreaks within a stanza. By default, it's set equal `\clubpenalty` at `\begin{document}` and thanks to the same trick as above you can set it also in the preamble.

`\verseskipbefore`   The default value both of `\verseskipbefore` and `\verseskipafter` is `\medskipamount`.
`\verseskipafter`

Four flavours of optical centering

The gmverse package provides optical centering of verse. By 'optical centering' I mean such a placing of the box containing a verse that it seems be balanced on the vertical axis of the page.

The optical centering may be automatic or manual. The automatic comes in four flavours, all of them cases of a general formula. The idea is to compute some average of the lines' lengths and set the rectangle's width to that average.

The first kind of average one can think of is arithmetical mean. That is the case W_1. The last kind of average is 'only the longest line counts', that's W_∞. Between them are c possibilities: consider a formula

$$W_\alpha = \sum_{k=1}^{n} l_k{}^\alpha \Big/ \sum_{k=1}^{n} l_k{}^{\alpha-1}$$

where l_k, $k = 1, \ldots n$ are the lengths of the lines[2] and $\alpha \in [1, +\infty)$.

When $\alpha = 1$, all the lines have 'equal right to decide' about the mean. When α grows, the longer lines become 'równiejsze' ('more equal', a term from the PRL epoch to describe unjust privileges of the Party Aparatchiks), which means they mean more to the mean.

We define W_∞ most naturally, as the limit:

$$W_\infty = \lim_{\alpha \to \infty} W_\alpha$$

and we notice easily that W_α corresponds with taking only the longest line into account.

Me personally like W_3 most and that's the default when optical centering is on.

If you prefer to set the centering manually, you are given two ways to do that: the `\vocpussyhair` parameter (dimen) intended for a slight modifications of the result of automatical computation, and the declaration(s)

`\versecenterdue{⟨benchmark text⟩}`

or

`\versecenterdue*{⟨benchmark dimen⟩}`.

New environments

verse0   You can define new verse environments in a way verseo is defined in line , that is in such definitions write `\verse` and `\endverse` and not `\begin{verse}` … `\end{%verse}`.

The Code

Package options

\ifverseloc@l	252	`\newif\ifverseloc@l`
local	253	`\DeclareOption{local}{\verseloc@ltrue}`
\ifsqu@rebr	255	`\newif\ifsqu@rebr`
squarebr	256	`\DeclareOption{squarebr}{\squ@rebrtrue}`
\ifv@c	258	`\newif\ifv@c`
voc	259	`\DeclareOption{voc}{\v@ctrue}`
vocweigh1	261	`\DeclareOption{vocweigh1}{\v@ctrue\AtEndOfPackage{\vocweigh1␣}}`
vocweigh2	262	`\DeclareOption{vocweigh2}{\v@ctrue\AtEndOfPackage{\vocweigh2␣}}`
vocweigh3	263	`\DeclareOption{vocweigh3}{\v@ctrue\AtEndOfPackage{\vocweigh3␣}}`
vocweigh8	264	`\DeclareOption{vocweigh8}{\v@ctrue\AtEndOfPackage{\vocweigh8␣}}`
	267	`\ProcessOptions`

Preliminaries and parameters

	271	`\long\def\firstofone#1{#1}`
	272	`\bgroup\catcode`\^^M=\active%`
	273	`\firstofone{\egroup%`
\defobeylines	274	`\def\defobeylines{\catcode`\^^M=\active␣\def^^M{\par}}}%`
\afterfi	276	`\long\def\afterfi#1#2\fi{\fi#1}`
\everyverse	278	`\newtoks\everyverse`
	279	`\everyverse={}`
\verseleftskip	280	`\newdimen\verseleftskip`
	281	`\verseleftskip=1,5em\relax`
\versealtleftskip	283	`\newlength{\versealtleftskip}`
	284	`\versealtleftskip=1,5em\relax`
\begversesquareh@ck	286	`\newcommand*\begversesquareh@ck{\relax}`
\parversesquareh@ck	287	`\newcommand*\parversesquareh@ck{\relax}`
\eversesquareh@ck	288	`\newcommand*\eversesquareh@ck{\relax}`
verselinenum	289	`\newcounter{verselinenum}`
\verseinstanzaparskip	290	`\newskip\verseinstanzaparskip`
\stanzaskip	293	`\@ifundefined{stanzaskip}{\newlength\stanzaskip}{}`
	294	`\stanzaskip=\medskipamount`
\verseskipbefore	296	`\newlength\verseskipbefore`
\verseskipafter	297	`\newlength\verseskipafter`
	298	`\verseskipbefore=\medskipamount`
	299	`\verseskipafter=\medskipamount`
\instanzapenalty	301	`\newcount\instanzapenalty␣\instanzapenalty=\maxdimen`
	302	`\AtBeginDocument{%`
	303	`\ifnum\instanzapenalty=\maxdimen`
	304	`\instanzapenalty=\clubpenalty`
	305	`\fi}`
\betweenstanzaspenalty	307	`\newcount\betweenstanzaspenalty␣\betweenstanzaspenalty=\maxdimen`
	308	`\AtBeginDocument{%`

---

[1] Unless you don't want it to be `\maxdimen`.

[2] Assume they all have nonzero length when $\alpha = 1$.

```
309    \ifnum\betweenstanzaspenalty=\maxdimen
310      \betweenstanzaspenalty=100
311    \fi}
```

Primarily the five parameters above were set \AtBeginDocument unconditionally ,
but it was not good, 'cause thus they couldn't be set other way in the preamble. As you
see, we use the fact that in some circumstances a \dimen register works as a \count.

Long line rest alignment to the right

```
322  \newcounter{stanzanum}
323  \newif\ifalterstanzas

326  \newif\if@linetogether
328  \@linetogethertrue

330  \newcommand*\versehangright{%
331  \renewenvironment{verse}{%
332    \par␣\setcounter{verselinenum}{0}\relax
```

As you'll see, stanza break is done by changing \parskip by \par in vmode and
then by \everypar. Below is preliminary—storing old value of \parskip. Idea of this
hack comes from Marcin Woliński.

```
336    \verseinstanzaparskip=\parskip\relax
337    \if@linetogether\interlinepenalty=\@M\fi
338    \clubpenalty=0\relax␣\widowpenalty=0\relax
339    \parindent=0pt\relax% here 'cause later optical centering deals with it.
340    \ifcenterverse␣\parindent=\linewidth
341        \advance␣\parindent␣by␣-\versegenericlinewidth
342        \advance␣\parindent␣by␣-0.5\parindent
343        \verseleftskip=\z@\relax% added 2006-07-15
344    \fi% of \ifcenterverse
345    \leftskip=\verseleftskip␣plus1fill\relax
346    \advance\leftskip␣by\vocpussyhair\relax
347     \ifmaxversewidth
348        \dimeno=\linewidth
349        \advance\dimeno␣by␣-\parindent
350        \ifdim\dimeno>\versemaxlinewidth
351            \rightskip=\linewidth
352            \advance\rightskip␣by-\parindent
353            \advance\rightskip␣by-\versemaxlinewidth
354            \advance\rightskip␣by0pt␣plus1fil
355            \else\rightskip=0pt␣plus1fil
356        \fi% of \ifdim
357        \else\rightskip=0pt␣plus1fil\relax
358     \fi% of \ifmaxversewidth
359    \parfillskip=0pt␣plus1fil␣\hyphenpenalty=100␣%
```

```
360    \newif\ifstanzabreak
361    \def\par{%
362      \ifvmode␣%
363        \ifnum0<\c@verselinenum\relax% we don't want stanza break before
                    verse
364          \ifstanzabreak
365            \stepcounter{stanzanum}%
366            \penalty10000␣\vfil\penalty\betweenstanzaspenalty%
                    \vfilneg\relax
```

As announced, we change \parskip to make a stanza break

```
368            \parskip=\stanzaskip\relax␣\stanzabreakfalse\relax
369            \ifalterstanzas
370              \ifodd\c@stanzanum
371                \ifnum\c@stanzanum>1\advance\leftskip-%
                          \versealtleftskip\fi
372              \else\advance\leftskip\versealtleftskip
373              \fi
374            \fi
375          \fi% of \ifstanzabreak
376        \else
377          \setcounter{stanzanum}{1}%
378        \fi% of \ifnum\c@verselinenum
379      \else
380        \penalty\instanzapenalty\relax\stanzabreaktrue\relax
381      \fi% of \ifvmode
382      \ifhmode␣\strut% to make square bracketing case work properly where
                      there'll be \nointerlineskip
384      \stepcounter{verselinenum}\relax
385      \endgraf\parversesquareh@ck\fi}
386    \everypar{% At new hmode we restore old value of \parskip.
387        \parskip=\verseinstanzaparskip\relax
```

and then we hack with stretchability

```
389        \hskipoptplus-1fill\relax}
```

At the beginning of verse we give some skip

```
391      \addvspace\verseskipbefore% not just \vskip 'cause this LATEX command will
                  check if there's another skip and 'll use the bigger one, not two.
```

\br@cketskip
```
394      \def\br@cketskip␣[####1]{\vspace{####1}}% four #s, indeed!
395      \def\\{\@ifstar{\nopagebreak\@ifnextchar{[}{\br@cketskip}{}}%
396        {\@ifnextchar{[}{\br@cketskip}{}}}%
397    \renewcommand*\newpage{\nopagebreak\vfil\pagebreak}%
```

I redefine \newpage because the original one spoils \leftskip or \rightskip, very important and delicate here.

```
401  \let\@doendpe\relax% so that \par redefing environments don't spoil verse's re-
                  def'ed \par
403  \def\^^M{\␣\par}% to make it work as in normal catcodes
405  \let\@sanitize\gmverse@sanitize% for \index macros in active
405  's
```

scope (see below)

```
407      \obeylines␣\the\everyverse␣\begversesquareh@ck%
408  }% of begdefverse enddef of verse:
410  {\ifhmode␣\par␣\fi␣\eversesquareh@ck%
413    \setbox0=\hbox{1232456789oqypf[]gp}\prevdepth=\dp0␣%
```

so that TEX deal with usual \baselineskip not add \lineskiplimit

```
415    \vskip\verseskipafter␣\relax
416    \global\vocpussyhair=\z@\relax
417    \@endpetrue
418  }% of enddef
419  }% of \versehangright
```

Long line rest begun with a [

The hacks given below i wrote with Marcin Woliński's help 2006/06/26

<div></div>

`\versehangrightsquare`   425 `\newcommand*\versehangrightsquare{\versehangright`

`\begversesquareh@ck`   426   `\renewcommand*\begversesquareh@ck{\relax\dimeno=\prevdepth`

427   `\setboxo=\vbox\bgroup\prevdepth=\dimeno␣\strut\e@tlineends}%`

`\parversesquareh@ck`   428   `\renewcommand*\parversesquareh@ck{\relax\ifnum\prevgraf>1`

429   `\setboxo=\lastbox␣␣\nointerlineskip%` not `\unskip`—we don't remove interline glue and set new but we assure there'll be a proper glue afrer first pass (see `\strut` before `\endgraf`) and don't add any new glue with [pass

435   `\hbox␣to␣\hsize{\hfill␣[\hskip-\leftskip\unhboxo}%`

436   `\fi\relax}`

`\eversesquareh@ck`   437   `\renewcommand*\eversesquareh@ck{\relax\egroup\unvboxo\relax}%`

438 `\squ@rebrtrue%` so that local calling of `\verseopticalcenter` call this redefinition if called after it

440 `}`

> The pair of macros below is used to gobble

442 `'s␣in␣|\obeylines|'`

> scope. It uses the fact that Knuthian `\obeylines` `\lets` not `\defs`.

`\e@tlineends`   444 `\def\e@tlineends{\futurelet\tokennotyete@ten\e@tlineend}`

`\e@tlineend`   445 `\def\e@tlineend{\ifx\par\tokennotyete@ten`

446   `\expandafter\expandafter\expandafter\e@tlineends\expandafter%`
     `\@gobble`

447   `\fi}`

<div></div>

Centering and optical centering

<div></div>

`\versegenericlinewidth`   452 `\newlength\versegenericlinewidth`

`\versemaxlinewidth`   453 `\newlength\versemaxlinewidth`

`\versepussyhair`   454 `\newlength\versepussyhair%` for additional correction in case of automatic optical centering.

`\ifcenterverse`   457 `\newif\ifcenterverse`

`\versecenterduest@r`   458 `\newcommand*\versecenterduest@r[1]{%`

459   `\versegenericlinewidth=#1\relax␣␣\centerversetrue}`

`\versecenterduest@rless`   460 `\newcommand*\versecenterduest@rless[1]{`

461   `\settowidth{\versegenericlinewidth}{#1}%`

462   `\centerversetrue}`

`\versecenterdue`   463 `\newcommand*\versecenterdue{%`

464   `\@ifstar{\versecenterduest@r}{\versecenterduest@rless}}`

`\ifmaxversewidth`   466 `\newif\ifmaxversewidth`

`\versemaxlinest@r`   467 `\newcommand*\versemaxlinest@r[1]{%`

468   `\versemaxlinewidth=#1\relax}`

`\versemaxlinest@rless`   469 `\newcommand*\versemaxlinest@rless[1]{%`

470   `\settowidth{\versemaxlinewidth}{#1}%`

471   `\maxversewidthtrue}`

`\versemaxline`   472 `\newcommand*\versemaxline{%`

473   `\@ifstar{\versemaxlinest@r}{\versemaxlinest@rless}}`

`\re@ltextw`   477 `\newdimen\re@ltextw%` to store the real `\textwidth`

`\vocpussyhair`   478 `\newdimen\vocpussyhair%` for a finishing touch after optical centering. The name is owed to A.T. (one of Polish typography Gurus), who used to say "Ja bym

to jeszcze przesunął o piczny włos w lewo" ('I'd move it left a pussy hair')
As you'll see, this register is cleared after each verse so you don't care about
scopes.

\gmvs@counta ⁴⁸⁵ `\newcount\gmvs@counta`

A scratch count for global assignments (originally it was `\count1`, not a good idea.

\verseoptic@lcenter ⁴⁸⁹ `\newcommand*\verseoptic@lcenter{%`
⁴⁹⁰ `  {\ifsqu@rebr\versehangrightsquare%`
⁴⁹¹ `    \else\versehangright\fi%`
⁴⁹² `    \global\let\inn@verse=\verse%`
⁴⁹³ `    \global\let\endinn@verse=\endverse\relax}%`

verse ⁴⁹⁵ `\renewenvironment{verse}{\par%`
\verse@contens ⁴⁹⁶ `  \newtoks\verse@contens␣\verse@contens={}\relax%`
\per@verse ⁴⁹⁷ `  \long\def\per@verse####1\end####2{%`
⁴⁹⁸ `    \addto@hook\verse@contens{####1}\checkifpr@perend{%`
`    ####2}}%`
\checkifpr@perend ⁴⁹⁹ `  \def\checkifpr@perend####1{\def\pr@perend@test{####1}%`
\pr@perend@test ⁵⁰⁰ `    \ifx\pr@perend@test\@currenvir%`
⁵⁰¹ `    \addto@hook\verse@contens{\par}\expandafter\end{####1}%`
⁵⁰² `    \else`
⁵⁰³ `    \addto@hook\verse@contens{\end{####1}}\expandafter%`
`    \per@verse%`
⁵⁰⁴ `  \fi}%`
⁵⁰⁵ `  \defobeylines\per@verse}%` of begdef So we have all the environment's con-
tents in the `\verse@contens` tokslist

⁵⁰⁸ `  {%`
⁵⁰⁹ `    {%` we switch off some commands
\br@cketskip ⁵¹⁰ `    \def\br@cketskip␣[####1]{}%` parameters are not allowed in enddef
⁵¹¹ `    \def\\{\@ifstar{\@ifnextchar{[}{\br@cketskip}{}}%`
⁵¹² `      {\@ifnextchar{[}{\br@cketskip}{}}}%`
⁵¹³ `    \renewcommand*\newpage{}%`

and set `\par` for the first boxing (to measure)

⁵¹⁵ `    \def\par{%`
⁵¹⁶ `      \ifhmode\endgraf%`
⁵¹⁷ `        \ifnum\prevgraf=1␣%`
⁵¹⁸ `          \setbox0=\lastbox\setbox2=\hbox{\unhbox0}%`
⁵¹⁹ `        \else`
⁵²⁰ `          \setbox2=\hbox␣to\re@ltextw{}%`
⁵²¹ `        \fi%`

If the last paragraph typeset has only one line, we give it the weighing subroutine.
If that paragraph has another number of lines, which means the respective line of the
verse is longer than `\textwidth`, then we give a `\textwidth`-long box the weighing
subroutine.

The question is still open how to center a verse with very long lines. The solution
chosen here seems to be reasonable if the long lines are rather seldom. (Note that weigh-
ing full-length lines `\prevgraf` − 1 times and then the last paragraph line isn't advis-
able since the line rest is aligned right so it rather acts as a full-length in perception
of the verse.) Note that the weighing subroutine always gets a line not longer than
`\textwidth`.

⁵³⁶ `        \voc@weighline␣2␣1␣\gmvs@counta␣%` parameters: #1 box reg-
ister, #2 dimen for total length, #3 count for sum of weights

8

538   `\fi`% of outermost `\ifhmode` and of `\par`

539   `}`% of `\def\par`

the very boxing and weighing preliminaries

543   `\re@ltextw=\textwidth\relax`

544   `\ifvoccubic\global\voc@edivs=0\relax\fi`% in cubic case we are
        quite close arithmetics' limitations. This count is for number of
        emergency divisions to avoid overflow.

547   `\textwidth=0.5\maxdimen\relax`%

548   `\global\gmvs@counta=0\relax\global\dimen1=\z@\relax`%

they'll be the number of lines and their scaled total length

550   `\setbox0=\vbox{`%

551   `\the\verse@contens}`% of `\vbox`

And the finale of weighing:

557   `\voc@weightotal␣1␣\gmvs@counta`% #1 input/result dimen, #2 input
        count.
        If there were emergency divisions just restore original `\voc@scale`
        and nullize `\voc@edivs`

560   `\ifnum\voc@edivs>0\global\voc@scale=\@@voc@scale\relax`

561   `\global\voc@edivs=0\relax`

562   `\fi`% of `\ifnum`

Common sense's touch:

564   `\ifdim\dimen1<\z@\global\dimen1=\z@\relax\fi`

565   `\ifdim\dimen1>\re@ltextw\global\dimen1=\re@ltextw\relax`%
       `\fi`

567   `}`% of group for first boxing

568   `\versecenterdue*{\dimen1}`%

569   `\global\dimen1=\z@\relax`%

and the very verse at last

571   `\expandafter\inn@verse`%

so that `\inn@verse` could see leading `^^M` signs

573   `\the\verse@contens`%

574   `\endinn@verse`%

575   `}`% of verse's enddef

576   `}`% of `\verseoptic@lcenter`

`\ifvocgamma` 579   `\newif\ifvocgamma`% to be able to make longer lines weigh more than short ones
    in the average.

`\ifvoccubic` 581   `\newif\ifvoccubic`% switch for cubic weight function.

582   `\voccubictrue`

`\ifvocmax` 583   `\newif\ifvocmax`% switch for max weight function

Two parameters below were experimentally set to given values under pdfε-TeX. The trial text was Wisława Szymborska's „Pod jedną gwiazdką" verse typed in one verse environment 11 times (and the cubic weighing option was taken) Chosen verse has many lines and all are long. If someone writes text with lines much than those more than 2 times, she should consider if what he writes is really a verse and therefore if should it be typeset in a verse environment.

`\voc@scale` 594   `\newcount\voc@scale␣\global\voc@scale="2000␣`% it'll give us .125pt preci-
    sion

\@@voc@scale	595 \def\@@voc@scale{"2000}
\vocunitlength	596 \newdimen\vocunitlength␣\global\vocunitlength=8pt␣%
\voc@edivs	597 \newcount\voc@edivs% voc emergency divisions
\defvocgamma	599 \newcommand*\defvocgamma[1][false]{%
	600 \csname␣vocgamma#1\endcsname
	601 \ifvocgamma
\voc@weighline	602 \def\voc@weighline##1##2##3{%
	% ##1 box register,
	% ##2 dimen for total length,
	% ##3 count for sum of weights

The box ##1 is \re@ltextw-wide at most, see the remark after line 521.

611 \dimen0=\wd##1␣%

we divide the length of a line by \vocunitlength

613 \dimen2=\vocunitlength
615 \count0=\dimen0␣%
616 \divide\count0␣by\dimen2␣% and advance ##3 by (square of) it.
618 \ifvoccubic\multiply\count0␣by␣\count0␣\fi
619 \global\advance##3by\count0␣%

Now we scale the line length and add it's multiple to the total scaled length.

622 \divide\dimen0␣by\voc@scale
623 \multiply\dimen0␣by\count0␣%
624 \global\advance\dimen##2by\dimen0␣%
625 \ifvoccubic% check if not danger of overflow and maybe emerg. divide
626 \ifdim\dimen##2>0.875\maxdimen
627 \global\divide\dimen##2␣by2␣%
628 \global\multiply\voc@scale␣by2␣%
629 \global\advance\voc@edivs␣by1␣%
630 \dimen8=\voc@scale␣sp\relax
631 \PackageWarningNoLine{gmverse}{%
632 To␣avoid␣overflow␣\on@line\MessageBreak
633 I␣had␣to␣diminish␣precision␣of␣optical␣centering.%
\MessageBreak
634 Current␣precision␣is␣\the\dimen8.\MessageBreak
635 (so␣far␣I␣have␣done␣\the\voc@edivs\space␣such␣
emergency
636 divisions.)}%
637 \fi% of ifdim
638 \fi% of \ifvoccubic
639 }% of \voc@weighline

\voc@weightotal	641 \def\voc@weightotal##1##2{%
	% ##1 input/result dimen,
	% ##2 input count

645 \ifnum##2>0\global\divide\dimen##1␣by##2␣\fi%
646 \global\multiply\dimen##1␣by\voc@scale}%

If we wish to use $_\infty$ weight function:

648 \ifvocmax

\voc@weighline	649 \def\voc@weighline##1##2##3{%
	% ##1 box register,
	% ##2 dimen for total length,

% ##3 count for sum of weights

The box ##1 is \re@ltextw-wide at most, see the remark after line 521.

657   \dimeno=\wd##1 %
658   \divide\dimeno by\voc@scale
659   \ifdim\dimeno>\dimen##2\global\dimen##2=\dimeno \fi
660   \global##3=1\relax
661   }% of \def\voc@weighline
662   \fi% of ifvocmax
663   \else% (= vocgammafalse)
\voc@weighline 664   \def\voc@weighline##1##2##3{%
                           % ##1 box register,
                           % ##2 dimen for total length,
                           % ##3 count for sum of weights
669   \global\advance##3by1\relax%

The box ##1 is \re@ltextw-wide at most, see the remark after line 521.

672   \dimeno=\wd##1 %
673   \divide\dimenoby\voc@scale
674   \global\advance\dimen##2by\dimeno }%
\voc@weightotal 675   \def\voc@weightotal##1##2{%
                           % ##1 input/result dimen,
                           % ##2 input count.
679   \ifnum\count##1>0\global\divide\dimen##1 by##2 \fi%
680   \global\multiply\dimen##1 by\voc@scale}%
681   \fi% of \ifvocgamma
682   }% of \defvocgamma

\verseopticalcenter 686 \def\verseopticalcenter{\@ifnextchar{[}{\voce@tbrackets}%
                     687   {\verseoptic@lcenter}}
\voce@tbrackets   688 \def\voce@tbrackets[#1]{\squ@rebrtrue\verseoptic@lcenter}

As you see, the \verseopticalcenter declaration has an optional parameter: if
you write (anything in) [] after it, it shall call squarebracket verse'ion. Anyway, if you
call it where \versehangrightsquare declaration is at work, \verseopticalcenter
will work with square bracket.

From my personal gedlyr package, to make possible put some prose into a verse i.e.,
justified paragraphs with no indent, separated by \stanzaskip:

\justrskip 700 \newlength{\justrskip}

\justrskip should be assigned in the very document to store value of \rightskip
out of verse. We assign nothing, so by default it's \z@.

prosato 704 \newenvironment*{prosato}{\catcode`\^^M=5\relax%
         705   \rightskip=\justrskip\relax%
         706   \dimeno\leftskip\relax%
         707   \leftskip\dimeno\relax%
         708   \parfillskip\z@ plus1fil\relax%
         709   \def\par{\ifhmode\stepcounter{verselinenum}\fi\endgraf}%
         710   \interlinepenalty=10\relax%
         711   \linepenalty=10\relax%
         712   \clubpenalty=10000 \widowpenalty=10000%
         713   \everypar{\parskip1\stanzaskip%
         714   \everypar{}%
         715 }%

```
716   \let\itm\relax% back compatibility
717   \let\\\newline%
718   \let\itl\\% back compatibility
720 }% of begdef.
721 {\par}% enddef.
```
You can still use \prosato as a command (a declaration).

\psalmato
```
725 \newcommand*\psalmato{\alterstanzastrue
726   \stanzaskiposp␣pluso,5ex\relax\ignoreactiveM}
```
last hooking

```
731 \ifverseloc@l\else
732 \ifsqu@rebr\AtBeginDocument{\versehangrightsquare}
733   \else\AtBeginDocument{\versehangright}
734   \fi% of \ifsqu@rebr
735   \ifv@c\AtBeginDocument{\defvocgamma[true]\centerversetrue%
         \verseoptic@lcenter}\fi
736   \fi
```

\gmverse@sanitize
```
740 \def\gmverse@sanitize{\@makeother\␣\@makeother\\\@makeother\$%
         \@makeother\&%
741   \@makeother\#\@makeother\^\@makeother\_%%\@makeother\% other% made
         me some trouble in \indexing long texts in \obeyllines' scope.
744 \@makeother\~%
745 \catcode`\^^M=5\relax}
```

```
748 \bgroup\obeylines
749 \firstofone{\egroup
```
\ignoreactiveM
```
750   \long\def\ignoreactiveM#1{%
751     \ifx
752     #1\afterfi\ignoreactiveM%
753     \else\afterfi#1\fi}}
```

verse0
```
756 \newenvironment*{verse0}
758 {\verseskipbefore\z@skip\verseskipafter\z@skip\verse}
759 {\endverse}
```
For easier declaring the degree of the weighing function:

\vocweigh
```
764 \newcommand*\vocweigh[2][\newcommand]{%
765   \ifx\newcommand#1\relax\afterfi\verseopticalcenter
766   \else\afterfi{\verseopticalcenter[]}%
767   \fi
768   \ifcase#2%
769   \or% 1
770     \afterfi{\voccubicfalse\vocmaxfalse\defvocgamma\relax}%
771   \or% 2
772     \afterfi{\voccubicfalse\vocmaxfalse\defvocgamma[true]}%
773   \or% 3
774     \afterfi{\voccubictrue\vocmaxfalse\defvocgamma[true]}
775   \else
776     \afterfi{\vocmaxtrue\voccubicfalse\defvocgamma[true]}%
777   \fi}
```

```
780 \endinput
```

Change History

v0.68
    `\if@linetogether`:
      added to meet ls-R needs, 326
v0.69
    General:
      Global use of `\count1` replaced with
      `\gmvs@counta` in `\voc@weighline`
      and `\voc@weightotal`, 780
v0.71
    General:

      CheckSum 825, 0
v0.72
    General:
      CheckSum 823, 0
v0.73
    General:
      A bug fix: obsolete `\afterelsefi`
      changed to `\afterfi`, 780
      CheckSum 822, 0